Evaluation of Text Generation: A Survey

Asli Celikyilmaz[*]
Microsoft Research
aslicel@microsoft.com

Elizabeth Clark[*]
University of Washington
eaclark7@cs.washington.edu

Jianfeng Gao[*]
Microsoft Research
jfgao@microsoft.com

Abstract

The paper surveys evaluation methods of natural language generation (NLG) systems that have been developed in the last few years. We group NLG evaluation methods into three categories: (1) human-centric evaluation metrics, (2) automatic metrics that require no training, and (3) machine-learned metrics. For each category, we discuss the progress that has been made and the challenges still being faced, with a focus on the evaluation of recently proposed NLG tasks and neural NLG models. We then present two case studies of automatic text summarization and long text generation, and conclude the paper by proposing future research directions.[2]

[*] Equal contribution.

[2] We are grateful for the following people: Rahul Jha, Sudha Rao, Ricky Lyond for their helpful comments and suggestions on earlier versions of this paper. We would like to thank the authors of the papers who gave us permission to use their figures, tables, and examples in our survey paper to summarize the related work.

Contents

Chapter 1

Introduction

Natural language generation (NLG), a sub-field of natural language processing (NLP), deals with building software systems that can produce coherent and readable text. NLG can be applied to a broad range of NLP tasks such as generating responses to user questions in a chatbot, translating a sentence or a document from one language into another, offering suggestions to help write a story, or generating summaries of time-intensive data analysis. NLG evaluation is challenging mainly because many NLG tasks are open-ended. For example, a dialog system can generate multiple plausible responses for the same user input. A document can be summarized in different ways. Therefore, human evaluation remains the gold standard for almost all NLG tasks. However, human evaluation is expensive, and researchers often resort to automatic metrics for quantifying day-to-day progress and for performing automatic system optimization. Recent advancements in deep learning have yielded tremendous improvements in many NLP tasks. This, in turn, presents a need for evaluating these deep neural network (DNN) models for NLG.

In this paper we provide a comprehensive survey of NLG evaluation methods with a focus on evaluating neural NLG systems. We group evaluation methods into three categories: (1) human-centric evaluation metrics, (2) automatic metrics that require no training, and (3) machine-learned metrics. For each category, we discuss the progress that has been made, the challenges still being faced, and proposals for new directions in NLG evaluation.

1.1 Evolution of Natural Language Generation

NLG is defined as the task of building software systems that can *write* (i.e., producing explanations, summaries, narratives, etc.) in English and other human languages[1]. Just as people communicate ideas through writing or speech, NLG systems are designed to produce natural language text or speech that conveys ideas to its readers in a clear and useful way. NLG systems have been used to generate text for many real-world applications such as generating weather forecasts, carrying interactive conversations with humans in spoken dialog systems (chatbots), captioning images or visual scenes, translating text from one language to another, and generating stories and news articles.

NLG techniques range from simple template-based systems that generate natural language text using rules and templates to machine-learned systems that have a complex understanding of human grammar. The first generation of automatic NLG systems uses rule-based or data-driven pipeline methods. In their seminal paper, Reiter & Dale (2000) present a classical three-stage NLG architecture, as shown in Figure 1.1. The first stage is *document planning*, in which the content and its order are determined and a text plan that outlines the structure of messages is generated. The second is the *micro-planning* stage, in which referring expressions that identify objects like entities or places are generated, along with the choice of words to be used and how they are aggregated. Collating similar sentences to improve readability with a natural flow also occurs in this stage. The last stage is *realization*, in which the actual text is generated, using linguistic knowledge about morphology, syntax, semantics, etc. Earlier work has focused on modeling discourse structures and learning representations of relations between text units for text generation (McKeown, 1985; Marcu, 1997;

[1] From Ehud Reiter's Blog (Reiter, 2019).

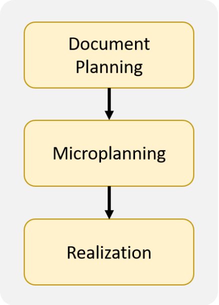


Figure 1.1: The three stages of the traditional NLG process (Reiter & Dale, 2000).

Ono et al., 1994; Stede & Umbach, 1998), for example using Rhetorical Structure Theory (Mann & Thompson, 1987) or Discourse Representation Theory (Lascarides & Asher, 1991). There is a large body of work that is based on template-based models and have used statistical methods to improve generation by introducing new methods such as sentence compression, reordering, lexical paraphrasing, and syntactic transformation, to name a few (Sporleder, 2005; Steinberger, 2006; Knight, 2000; Clarke & Lapata, 2008; Quirk et al., 2004).

These earlier text generation approaches and their extensions play an important role in the evolution of NLG research. The same is true for the NLG research in the last decade, in which we witness a paradigm shift towards learning representations from large textual corpora in an unsupervised manner using deep neural network (DNN) models. Recent NLG models are built by training DNN models, typically on very large corpora of human-written texts. The paradigm shift starts with the use of recurrent neural networks (Graves, 2013) (e.g., long-short term memory networks (LSTM) (Hochreiter & Schmidhuber, 1997), gated recurrent units (GRUs) (Cho et al., 2014), etc.) for learning language representations, such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), and later sequence-to-sequence learning (Sutskever et al., 2014), which opens up a new chapter characterised by the wide application of the encoder-decoder architecture. Although sequence-to-sequence models were originally developed for machine translation they were soon shown to improve performance across many NLG tasks. These models' weakness of capturing long-span dependencies in long word sequences motivates the development of *attention networks* (Bahdanau et al., 2015) and *pointer networks* (Vinyals et al., 2015). The Transformer architecture (Vaswani et al., 2017), which incorporates an encoder and a decoder, both implemented using the self-attention mechanism, is being adopted by new state-of-the-art NLG systems. There has been a large body of research in recent years that focuses on improving the performance of NLG using large-scale pre-trained language models for contextual word embeddings (Peters et al., 2018; Devlin et al., 2018; Sun et al., 2019; Dong et al., 2019), using better sampling methods to reduce degeneration in decoding (Zellers et al., 2019; Holtzman et al., 2020), and learning to generate text with better discourse structures and narrative flow (Yao et al., 2018; Fan et al., 2019b; Dathathri et al., 2020; Rashkin et al., 2020).

Neural models have been applied to many NLG tasks which we will discuss in this paper, including:

- summarization: common tasks include single or multi-document tasks, query-focused or generic summarization, and summarization of news, meetings, screen-plays, social blogs, etc.
- machine translation: sentence- or document-level.
- dialog response generation: goal-oriented or chit-chat dialogs.
- paraphrasing
- question generation
- long text generation: most common tasks are story, news, or poem generation.
- data-to-text generation: e.g., table summarization.
- caption generation from non-text input: input can be tables, images, or sequences of video frames (e.g., in visual storytelling), to name a few.

1.2 Why a Survey on Evaluation on Natural Language Generation

The question we are interested in in this paper is how to measure the quality of text generated from NLG models.

Text generation is a key component of language translation, chatbots, question answering, summarization, and several other applications that people interact with everyday. Building language models using traditional approaches is a complicated task that needs to take into account multiple aspects of language, including linguistic structure, grammar, word usage, and perception, and thus requires non-trivial data labeling efforts. Recently, Transformer-based neural language models have shown very effective in leveraging large amounts of raw text corpora from online sources (such as Wikipedia, search results, blogs, Reddit posts, etc.). For example, one of most advanced neural language models, GPT-2 (Radford et al., 2019), can generate long texts that are almost indistinguishable from human-generated texts (Zellers et al., 2019). Empathetic social chatbots, such as XiaoIce (Zhou et al., 2020), seem to understand human dialog well and can generate interpersonal responses to establish long-term emotional connections with users.

Nevertheless, training a powerful language model relies on evaluation metrics that can measure the model quality from different perspectives. For instance, it is imperative to build evaluation methods that can determine whether a text is generated by a human or a machine to prevent any potential harm. Similarly, evaluating the generated text based on factual consistency has recently drawn attention in the NLG field. It is concerning that neural language models can generate open-ended texts that are fluent but not grounded in real-world knowledge or facts, such as fake news. The situation is particularly alarming if the generated reports or news are related to the well-being of humankind, such as summaries of health reports (Zhang et al., 2019b). Thus, in addition to mainstream NLG evaluation methods, our survey also discusses recently proposed metrics to address human-facing issues, such as the metrics that evaluate the factual consistency of a generated summary or the empathy level of a chatbot's response.

Many NLG surveys have been published in the last few years (Gatt & Krahmer, 2017; Zhu et al., 2018; Zhang et al., 2019a). Others survey specific NLG tasks or NLG models, such as image captioning (Kilickaya et al., 2017; Hossain et al., 2018; Li et al., 2019; Bai & An, 2018), machine translation (Dabre et al., 2020; Han & Wong, 2016; Wong & Kit, 2019), summarization (Deriu et al., 2009; Shi et al., 2018), question generation (Pan et al., 2019), extractive key-phrase generation (Çano & Bojar, 2019), deep generative models (Pelsmaeker & Aziz, 2019; Kim et al., 2018), text-to-image synthesis (Agnese et al., 2020), and dialog response generation (Liu et al., 2016; Novikova et al., 2017; Deriu et al., 2019; Dusek et al., 2019; Gao et al., 2019), to name a few.

There are only a few published papers that review evaluation methods for specific NLG tasks, such as image captioning (Kilickaya et al., 2017), machine translation (Goutte, 2006), online review generation (Garbacea et al., 2019), interactive systems (Hastie & Belz, 2014a), and conversational dialog systems (Deriu et al., 2019), and for human-centric evaluations (Lee et al., 2019; Amidei et al., 2019b). The closest to our paper is the NLG survey paper of Gkatzia & Mahamood (2015), which includes a chapter on NLG evaluation metrics.

Different from this work, our survey is dedicated to NLG evaluation, with a focus on the evaluation metrics developed recently for neural text generation systems, and provides an in-depth analysis of existing metrics to-date. To the best of our knowledge, our paper is the most extensive and up-to-date survey on NLG evaluation.

1.3 Outline of The Survey

We review NLG evaluation methods in three categories in Chapters 2-4:

- **Human-Centric Evaluation.** The most natural way to evaluate the quality of a text generator is to involve *humans as judges*. Naive or expert subjects are asked to rate or compare texts generated by different NLG systems or to perform a Turing test (Turing, 1950) to distinguish machine-generated texts from human-generated texts. Most human evaluations are task-specific, and thus need to be designed and implemented differently for the outputs of different tasks. For example, the human evaluation for image captioning is different from one for text summarization.

- **Untrained Automatic Metrics.** This category, also known as *automatic metrics*, is the most commonly used in the research community. These evaluation methods compare machine-generated texts to human-generated texts (references) from the same input data using metrics that do not require machine learning but are simply based on string overlap, content overlap, string distance, or lexical diversity, such as n-gram match and distribution similarity. For most NLG tasks, it is critical to select the right automatic metric that measures the aspects of the generated text that are consistent with the original design goals of the NLG system.
- **Machine-Learned Metrics.** These metrics are often based on machine-learned models, which are used to measure the similarity between two machine-generated texts or between machine-generated and human-generated texts. These models can be viewed as digital judges that simulate human judges. We investigate the differences among these evaluations and shed light on the potential factors that contribute to these differences.

In Chapter 5, we present two case studies of evaluation methods developed for two tasks, automatic document summarization and long-text generation (e.g., story or review generation), respectively. We choose these tasks because they have attracted a lot of attention in the NLG research community and the task-specific evaluation metrics they used can be adopted for other NLG tasks. We then provide general guidelines in building evaluation metrics that correlate well with human judgements. Lastly, we conclude the paper with future research directions for NLG evaluation.

Chapter 2

Human-Centric Evaluation Methods

Whether a system is generating an answer to a user's query, a justification for a classification model's decision, or a short story, the ultimate goal in NLG is to generate text that is valuable to people. For this reason, human evaluations are typically viewed as the most important form of evaluation for NLG systems and are held as the gold standard when developing new automatic metrics. Since automatic metrics still fall short of replicating human decisions (Reiter & Belz, 2009b; Krahmer & Theune, 2010; Reiter, 2018), many NLG papers include some form of human evaluation. For example, Hashimoto et al. (2019) report that 20 out of 26 generation papers published at ACL2018 present human evaluation results.

While human evaluations give the best insight into how well a model performs in a task, it is worth noting that human evaluations also pose several challenges. First, human evaluations can be expensive and time-consuming to run, especially for the tasks that require extensive domain expertise. While online crowd-sourcing platforms such as Amazon Mechanical Turk have enabled researchers to run experiments on a larger scale at a lower cost, they come with their own problems, such as maintaining quality control (Ipeirotis et al., 2010; Mitra et al., 2015). Furthermore, even with a large group of annotators, there are some dimensions of generated text that are not well-suited to human evaluations, such as diversity (Hashimoto et al., 2019). There is also a lack of consistency in how human evaluations are run, which prevents researchers from reproducing experiments and comparing results across systems. This inconsistency in evaluation methods is made worse by inconsistent reporting on methods; details on how the human evaluations were run are often incomplete or vague. For example, van der Lee et al. (2019) find that in a sample of NLG papers from ACL and INLG, only 55% of papers report the number of participants in their human evaluations.

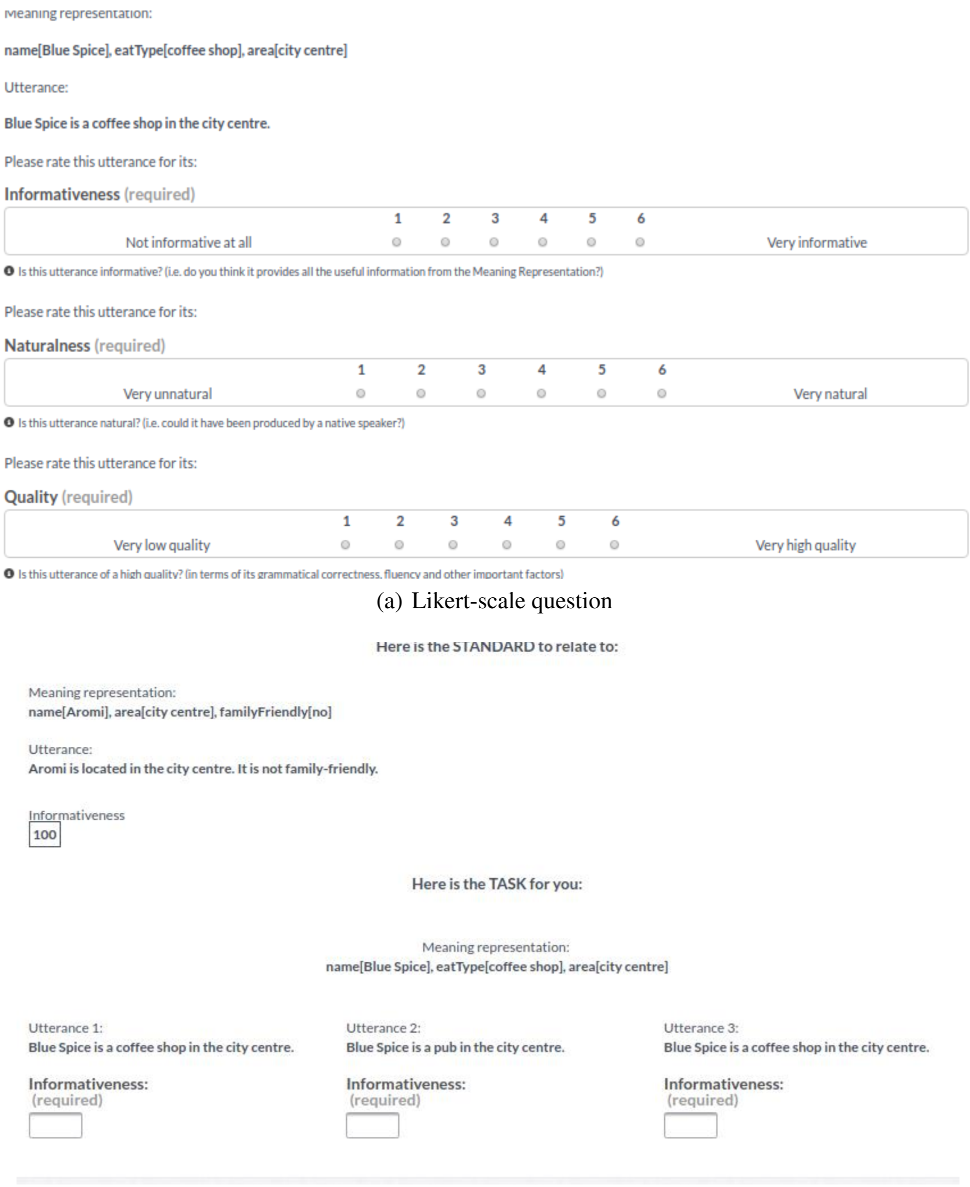
In this chapter, we describe common approaches researchers take when evaluating generated text using only human judgments, grouped into intrinsic (§2.1) and extrinsic (§2.2) evaluations (Belz & Reiter, 2006). However, there are other ways to incorporate human subjects into the evaluation process, such as training models on human judgments, which will be discussed in Chapter 4.

2.1 Intrinsic Evaluation

An *intrinsic evaluation* asks people to evaluate the quality of generated text, either overall or along some specific dimension (e.g., fluency, coherence, correctness, etc.). This is typically done by generating several samples of text from a model and asking human evaluators to score their quality.

The simplest way to get this type of evaluation is to show the evaluators the generated texts one at a time and have them judge their quality individually. They are asked to vote whether the text is good or bad, or to make more fine-grained decisions by marking the quality along a Likert or sliding scale (see Figure 2.1(a)). However, judgments in this format can be inconsistent and comparing these results is not straightforward; Amidei et al. (2019b) find that analysis on NLG evaluations in this format is often done incorrectly or with little justification for the chosen methods.

To more directly compare a model's output against baselines, model variants, or human-generated text, intrinsic evaluations can also be performed by having people choose which of two generated



Meaning representation:

name[Blue Spice], eatType[coffee shop], area[city centre]

Utterance:

Blue Spice is a coffee shop in the city centre.

Please rate this utterance for its:

Informativeness (required)

	1	2	3	4	5	6	
Not informative at all	○	○	○	○	○	○	Very informative

❶ Is this utterance informative? (i.e. do you think it provides all the useful information from the Meaning Representation?)

Please rate this utterance for its:

Naturalness (required)

	1	2	3	4	5	6	
Very unnatural	○	○	○	○	○	○	Very natural

❶ Is this utterance natural? (i.e. could it have been produced by a native speaker?)

Please rate this utterance for its:

Quality (required)

	1	2	3	4	5	6	
Very low quality	○	○	○	○	○	○	Very high quality

❶ Is this utterance of a high quality? (in terms of its grammatical correctness, fluency and other important factors)

(a) Likert-scale question

Here is the STANDARD to relate to:

Meaning representation:
name[Aromi], area[city centre], familyFriendly[no]

Utterance:
Aromi is located in the city centre. It is not family-friendly.

Informativeness
100

Here is the TASK for you:

Meaning representation:
name[Blue Spice], eatType[coffee shop], area[city centre]

Utterance 1:
Blue Spice is a coffee shop in the city centre.

Informativeness:
(required)

Utterance 2:
Blue Spice is a pub in the city centre.

Informativeness:
(required)

Utterance 3:
Blue Spice is a coffee shop in the city centre.

Informativeness:
(required)

(b) RankME-style question

Figure 2.1: Two different methods for obtaining intrinsic evaluations of text generated from a meaning representation. Image Source: (Novikova et al., 2018), `https://github.com/jeknov/RankME`

texts they prefer, or more generally, rank a set of generated texts. This comparative approach has been found to produce higher inter-annotator agreement (Callison-Burch et al., 2007) in some cases. However, while it captures models' relative quality, it does not give a sense of the absolute quality of the generated text. One way to address this is to use a method like RankME (Novikova et al., 2018), which adds magnitude estimation (Bard et al., 1996) to the ranking task, asking evaluators to indicate how much better their chosen text is over the alternative(s) (see Figure 2.1(b)). Comparison-based approaches can become prohibitively costly (by requiring lots of head-to-head comparisons) or complex (by requiring participants to rank long lists of output) when there are many models to compare, though there are methods to help in these cases. For example, best-worst scaling (Louviere et al., 2015) has been used in NLG tasks (Kiritchenko & Mohammad, 2016; Koncel-Kedziorski et al., 2019) to simplify comparative evaluations; best-worst scaling asks participants to choose the best and worst elements from a set of candidates, a simpler task than fully ranking the set that still provides reliable results.

Almost all the text generation tasks today are evaluated with intrinsic human evaluations. Machine translation is one of the text generation tasks in which intrinsic human evaluations have made a huge impact on the development of more reliable and accurate translation systems, as automatic metrics are validated through correlation with human judgments. One metric that is most commonly used to judge translated output by humans is measuring its *adequacy*, which is defined by the *Linguistic Data Consortium* as "how much of the meaning expressed in the gold-standard translation or source is also expressed in the target translation."[1]. The annotators must be bilingual in both the source and target languages in order to judge whether the information is preserved across translation. Another dimension of text quality commonly considered in machine translation is *fluency*, which measures the quality of the generated text only (e.g., the target translated sentence), without taking the source into account. It accounts for criteria such as grammar, spelling, choice of words, and style. A typical scale used to measure fluency is based on the question "Is the language in the output fluent?". Fluency is also adopted in several text generation tasks including document summarization (Celikyilmaz et al., 2018; Narayan et al., 2018), recipe generation (Bosselut et al., 2018), image captioning (Lan et al., 2017), video description generation (Park et al., 2018), and question generation (Du et al., 2017), to name a few.

While fluency and adequacy have become standard dimensions of human evaluation for machine translation, not all text generation tasks have an established set of dimensions that researchers use. Nevertheless, there are several dimensions that are common in human evaluations for generated text. As with adequacy, many of these dimensions focus on the contents of the generated text. *Factuality* is important in tasks that require the generated text to accurately reflect facts described in the context. For example, in tasks like data-to-text generation or summarization, the information in the output should not contradict the information in the input data table or news article. This is a challenge to many neural NLG models, which are known to "hallucinate" information (Holtzman et al., 2020; Welleck et al., 2019); Maynez et al. (2020) find that over 70% of generated single-sentence summaries contained hallucinations, a finding that held across several different modeling approaches. Even if there is no explicit set of facts to adhere to, researchers may want to know how well the generated text follows rules of *commonsense* or how *logical* it is. For generation tasks that involve extending a text, researchers may ask evaluators to gauge the *coherence* or *consistency* of a text—how well it fits the provided context. For example, in story generation, do the same characters appear throughout the generated text, and do the sequence of actions make sense given the plot so far?

Other dimensions focus not on what the generated text is saying, but how it is being said. As with fluency, these dimensions can often be evaluated without showing evaluators any context. This can be something as basic as checking for simple language errors by asking evaluators to rate how *grammatical* the generated text is. It can also involve asking about the overall *style*, *formality*, or *tone* of the generated text, which is particularly important in style-transfer tasks or in multi-task settings. Hashimoto et al. (2019) ask evaluators about the *typicality* of generated text; in other words, how often do you expect to see text that looks like this? These dimensions may also focus on how efficiently the generated text communicates its point by asking evaluators how *repetitive* or *redundant* it is.

[1] https://catalog.ldc.upenn.edu/docs/LDC2003T17/TransAssess02.pdf

Note that while these dimensions are common, they may be referred to by other names, explained to evaluators in different terms, or measured in different ways (van der Lee et al., 2019). More consistency in how user evaluations are run, especially for well-defined generation tasks, would be useful for producing comparable results and for focused efforts for improving performance in a given generation task. One way to enforce this consistency is by handing over the task of human evaluation from the individual researchers to an evaluation platform, usually run by people hosting a shared task or leaderboard. In this setting, researchers submit their models or model outputs to the evaluation platform, which organizes and runs all the human evaluations. For example, ChatEval is an evaluation platform for open-domain chatbots based on both human and automatic metrics (Sedoc et al., 2019), and TuringAdvice (Zellers et al., 2020) tests models' language understanding capabilities by having people read and rate the models' ability to generate advice. Of course, as with all leaderboards and evaluation platforms, with uniformity and consistency come rigidity and the possibility of overfitting to the wrong objectives. Thus, how to standardize human evaluations should take this into account. A person's goal when producing text can be nuanced and diverse, and the ways of evaluating text should reflect that.

2.2 Extrinsic Evaluation

An *extrinsic evaluation* has people evaluate a system's performance on the task for which it was designed. Extrinsic evaluations are the most meaningful evaluation as they show how a system actually performs in a downstream task, but they can also be expensive and difficult to run (Reiter & Belz, 2009a). For this reason, intrinsic evaluations are more common than extrinsic evaluations (Gkatzia & Mahamood, 2015; van der Lee et al., 2019) and have become increasingly so, which van der Lee et al. (2019) attribute to a recent shift in focus on NLG subtasks rather than full systems.

Extrinsic methods measure how successful the system is in a downstream task. This success can be measured from two different perspectives: a user's success in a task and the system's success in fulfilling its purpose (Hastie & Belz, 2014b). Extrinsic methods that measure a user's success at a task look at what the user is able to take away from the system, e.g., improved decision making, higher comprehension accuracy, etc. (Gkatzia & Mahamood, 2015). For example, Young (1999), which Reiter & Belz (2009a) point to as one of the first examples of extrinsic evaluation of generated text, evaluate automatically generated instructions by the number of mistakes subjects made when they followed them. System success extrinsic evaluations, on the other hand, measure an NLG system's ability to complete the task for which it has been designed. For example, Reiter et al. (2003) generate personalized smoking cessation letters and report how many recipients actually gave up smoking.

Extrinsic human evaluations are commonly used in evaluating the performance of dialog (Deriu et al., 2019) and have made an impact on the development of the dialog modeling systems. Various approaches have been used to measure the system's performance when talking to people, such as measuring the conversation length or asking people to rate the system. The feedback is collected by real users of the dialog system (Black et al., 2011; Lamel et al., 2000; Zhou et al., 2020) at the end of the conversation. The Alexa Prize[2] follows a similar strategy by letting real users interact with operational systems and gathering the user feedback over a span of several months. However, the most commonly used human evaluations of dialog systems is still via crowd-sourcing platforms such as Amazon Mechanical Turk (AMT) (Serban et al., 2016a; Peng et al., 2020; Li et al., 2020; Zhou et al., 2020). Jurcícek et al. (2011) suggest that using enough crowd-sourced users can yield a good quality metric, which is also comparable to the human evaluations in which subjects interact with the system and evaluate afterwards.

2.3 The Evaluators

For many NLG evaluation tasks, no specific expertise is required of the evaluators other than a proficiency in the language of the generated text. This is especially true when fluency-related aspects of the generated text are the focus of the evaluation. Often, the target audience of an NLG system is broad, e.g., a summarization system may want to generate text for anyone who is interested in

[2] `https://developer.amazon.com/alexaprize`

reading news articles or a chatbot needs to carry a conversation with anyone who could access it. In these cases, human evaluations benefit from being performed on as wide a population as possible.

Typically evaluations in these settings are performed either in-person or online. An in-person evaluation could simply be performed by the authors or a group of evaluators recruited by the researchers to come to the lab and participate in the study. The benefits of in-person evaluation are that it is easier to train and interact with participants, and that it is easier to get detailed feedback about the study and adapt it as needed. Researchers also have more certainty and control over who is participating in their study, which is especially important when trying to work with a more targeted set of evaluators. However, in-person studies can also be expensive and time-consuming to run. For these reasons, in-person evaluations tend to include fewer participants, and the set of people in proximity to the research group may not accurately reflect the full set of potential users of the system. In-person evaluations may also be more susceptible to response biases, adjusting their decisions to match what they believe to be the researchers' preferences or expectations (Nichols & Maner, 2008; Orne, 1962).

To mitigate some of the drawbacks of in-person studies, online evaluations of generated texts have become increasingly popular. While researchers could independently recruit participants online to work on their tasks, it is common to use crowdsourcing platforms that have their own users whom researchers can recruit to participate in their task, either by paying them a fee (e.g., Amazon Mechanical Turk[3]) or rewarding them by some other means (e.g., LabintheWild[4], which provides participants with personalized feedback or information based on their task results). These platforms allow researchers to perform large-scale evaluations in a time-efficient manner, and they are usually less expensive (or even free) to run. They also allow researchers to reach a wider range of evaluators than they would be able to recruit in-person (e.g., more geographical diversity). However, maintaining quality control online can be an issue (Ipeirotis et al., 2010; Oppenheimer et al., 2009), and the demographics of the evaluators may be heavily skewed depending on user base of the platform (Difallah et al., 2018; Reinecke & Gajos, 2015). Furthermore, there may be a disconnect between what evaluators online being paid to complete a task would want out of a NLG system and what the people who would be using the end product would want.

Not all NLG evaluation tasks can be performed by any subset of speakers of a given language. Some tasks may not transfer well to platforms like Amazon Mechanical Turk where workers are more accustomed to dealing with large batches of microtasks. Specialized groups of evaluators can be useful when testing a system for a particular set of users, as in extrinsic evaluation settings. Researchers can recruit people who would be potential users of the system, e.g., students for educational tools or doctors for bioNLP systems. Other cases that may require more specialized human evaluation are projects where evaluator expertise is important for the task or when the source texts or the generated texts consist of long documents or a collection of documents. Consider the task of citation generation (Luu et al., 2020): given two scientific documents A and B, the task is to generate a sentence in document A that appropriately cites document B. To rate the generated citations, the evaluator must be able to read and understand two different scientific documents and have general expert knowledge about the style and conventions of academic writing. For these reasons, Luu et al. (2020) choose to run human evaluations with expert annotators (in this case, NLP researchers) rather than regular crowdworkers.

2.4 Inter-Evaluator Agreement

While evaluators often undergo training to standardize their evaluations, evaluating generated natural language will always include some degree of subjectivity. Evaluators may disagree in their ratings, and the level of disagreement can be a useful measure to researchers. High levels of inter-evaluator agreement generally mean that the task is well-defined and the differences in the generated text are consistently noticeable to evaluators, while low agreement can indicate a poorly defined task or that there are not reliable differences in the generated text.

Nevertheless, measures of inter-evaluator agreement are not frequently included in NLG papers. Only 18% of the 135 generation papers reviewed in Amidei et al. (2019a) include agreement analysis (though on a positive note, it was more common in the most recent papers they studied). When agree-

[3] https://www.mturk.com/

[4] http://www.labinthewild.org/

ment measures are included, agreement is usually low in generated text evaluation tasks, lower than what is typically considered "acceptable" on most agreement scales (Amidei et al., 2018, 2019a). However, as Amidei et al. (2018) point out, given the richness and variety of natural language, pushing for the highest possible inter-annotator agreement may not be the right choice when it comes to NLG evaluation.

While there are many ways to capture the agreement between annotators (Banerjee et al., 1999), we highlight the most common approaches used in NLG evaluation. For an in-depth look at annotator agreement measures in natural language processing, refer to Artstein & Poesio (2008).

2.4.1 Percent agreement

A simple way to measure agreement is to report the percent of cases in which the evaluators agree with each other. If you are evaluating a set of generated texts X by having people assign a score to each text x_i, then let a_i be the agreement in the scores for x_i (where $a_i = 1$ if the evaluators agree and $a_i = 0$ if they don't). Then the percent agreement for the task is:

$$P_a = \frac{\sum_{i=0}^{|X|} a_i}{|X|} \tag{2.1}$$

So $P_a = 0$ means the evaluators did not agree on their scores for any generated text, while $P_a = 1$ means they agreed on all of them.

However, while this is a common way people evaluate agreement in NLG evaluations (Amidei et al., 2019a), it does not take into account the fact that the evaluators may agree purely by chance, particularly in cases where the number of scoring categories are low or some scoring categories are much more likely than others (Artstein & Poesio, 2008). We need a more complex agreement measure to capture this.

2.4.2 Cohen's κ

Cohen's κ (Cohen, 1960) is an agreement measure that can capture evaluator agreements that may happen by chance. In addition to P_a, we now consider P_c, the probability that the evaluators agree by chance. So, for example, if two evaluators (e_1 and e_2) are scoring texts X with a score from the set S, then P_c would be the odds of them both scoring a text the same:

$$P_c = \sum_{s \in S} P(s|e_1) * P(s|e_2) \tag{2.2}$$

For Cohen's κ, $P(s|e_i)$ is estimated using the frequency with which Evaluator e_i assigned each of the scores across the task.[5] So, for example, if there are two scores, 0 and 1, and e_1 assigns 6 scores as 0s and 4 scores as 1s, and e_2 assigns 5 0s and 5 1s, then $P_c = 0.6 * 0.5 + 0.4 * 0.5$.

Once we have both P_a and P_c, Cohen's κ can then be calculated as:

$$\kappa = \frac{P_a - P_c}{1 - P_c} \tag{2.3}$$

2.4.3 Fleiss' κ

As seen in Equation 2.2, Cohen's κ measures the agreement between two annotators, but often many evaluators have scored the generated texts, particularly in tasks that are run on crowdsourcing platforms. Fleiss' κ (Fleiss, 1971) can measure agreement between multiple evaluators. This is done by still looking at how often pairs of evaluators agree, but now considering all possible pairs

[5] There are other related agreement measures, e.g., Scott's π (Scott, 1955), that only differ from Cohen's κ in how to estimate $P(s|e_i)$. These are well described in Artstein & Poesio (2008), but we do not discuss these here as they are not commonly used for NLG evaluations (Amidei et al., 2019a).

of evaluators. So now a_i, which we defined earlier to be the agreement in the scores for a generated text x_i, is calculated across all evaluator pairs:

$$a_i = \frac{\sum_{s \in S} \text{\# of evaluator pairs who score } x_i \text{ as } s}{\text{total \# of evaluator pairs}} \tag{2.4}$$

Then we can once again define P_a, the overall agreement probability, as it is defined in Equation 2.1—the average agreement across all the texts.

To calculate P_c, we estimate the probability of a judgment $P(s|e_i)$ by the frequency of the score across all annotators and assuming each annotator is equally likely to draw randomly from this distribution. So if r_s is the proportion of judgments that assigned a score s, then the likelihood of two annotators assigning score s by chance is $r_s * r_s = r_s^2$. Then our overall probability of chance agreement is:

$$P_c = \sum_{s \in S} r_s^2 \tag{2.5}$$

With these values for P_a and P_c, we can use Equation 2.3 to calculate Fleiss' κ.

2.4.4 Krippendorff's α

Each of the above measures treats all evaluator disagreements as equally bad, but in some cases, we may wish to penalize some disagreements more harshly than others. Krippendorff's α (Krippendorff, 1970), which is technically a measure of evaluator *disagreement* rather than agreement, allows different levels of disagreement to be taken into account.[6]

Like the κ measures above, we again use the frequency of evaluator agreements and the odds of them agreeing by chance. However, we will now state everything in terms of disagreement. First, we find the probability of disagreement across all the different possible score pairs (s_m, s_n), which are weighted by whatever value $w_{m,n}$ we assign the pair. So:

$$P_d = \sum_{m=0}^{|S|} \sum_{n=0}^{|S|} w_{m,n} \sum_{i=0}^{|X|} \frac{\text{\# of evaluator pairs that assign } x_i \text{ as } (s_m, s_n)}{\text{total \# of evaluator pairs}} \tag{2.6}$$

(Note that when $m == n$, i.e., the pair of annotators agree, $w_{m,n}$ should be 0.)

Next, to calculate the expected disagreement, we make a similar assumption as in Fleiss' κ: the random likelihood of an evaluator assigning a score s_i can be estimated from the overall frequency of s_i. If $r_{m,n}$ is the proportion of all evaluation pairs that assign scores s_m and s_n, then we can treat it as the probability of two evaluators assigning scores s_m and s_n to a generated text at random. So P_c is now:

$$P_c = \sum_{m=0}^{|S|} \sum_{n=0}^{|S|} w_{m,n} r_{m,n} \tag{2.7}$$

Finally, we can calculate Krippendorff's α as:

$$\alpha = 1 - \frac{P_d}{P_c} \tag{2.8}$$

[6] Note that there are other measures that permit evaluator disagreements to be weighted differently. For example, weighted κ (Cohen, 1968) extends Cohen's κ by adding weights to each possible pair of score assignments. In NLG evaluation, though, Krippendorff's α is the most common of these weighted measures; in the set of NLG papers surveyed in Amidei et al. (2019a), only 1 used weighted κ.

Chapter 3

Untrained Automatic Evaluation Metrics

With the increase of the numbers of NLG applications and their benchmark datasets, evaluation of NLG systems has become increasingly important. Today, the best evaluation for automatic NLG system output is human-based evaluation. However, human evaluation is costly and time-consuming to design and run, and more importantly, the results are not always repeatable (Belz & Reiter, 2006). Thus, automatic evaluation metrics are employed as an alternative in both developing new models and comparing them against state-of-the-art. In this survey, we group automatic metrics into two categories: untrained automatic metrics that do not require training (this chapter), and machine-learned evaluation metrics that are based on machine-learned models (Chapter 4).

In this chapter we review untrained automatic metrics used in different NLG applications and discuss their advantages and drawbacks in comparison with other approaches. Untrained automatic metrics for NLG evaluation are used to measure the effectiveness of the models that generate text, such as in machine translation, image captioning, or question generation. These metrics compute a score that indicates the similarity (or dissimilarity) between an automatically generated text and human-written reference (gold standard) text. Untrained automatic evaluation metrics are fast and efficient and are widely used to quantify day-to-day progress of model development, e.g., comparing model training with different hyperparameters. We group the untrained automatic evaluation methods, as in Table 3.1, into five categories:

- *n*-gram overlap metrics
- distance-based metrics
- diversity metrics
- content overlap metrics
- grammatical feature based metrics

3.1 *n*-gram Overlap Metrics for Content Selection

n-gram overlap metrics are commonly used for evaluating NLG systems and measure the degree of "matching" between machine-generated and human-authored (ground-truth) texts. In this section we present several *n*-gram match features and the NLG tasks they are used to evaluate.

	Metric	Property	MT	IC	SR	SUM	DG	QG	RG
n-gram overlap	BLEU	*n*-gram precision	✓	✓			✓	✓	✓
	NIST	*n*-gram precision	✓						
	F-SCORE	precision and recall	✓	✓	✓	✓	✓	✓	✓
	WER	% of insert,delete,replace			✓				
	ROUGE	*n*-gram recall				✓	✓		
	METEOR	*n*-gram w/ synonym matching	✓	✓			✓		
	HLEPOR	unigrams harmonic mean	✓						
	RIBES	unigrams harmonic mean							
	CIDER	*tf-idf* weighted *n*-gram similarity		✓					
distance-based	EDIT DIST.	cosine similarity	✓	✓	✓	✓	✓	✓	
	TER	translation edit rate	✓						
	WMD	earth mover distance on words		✓		✓			
	SMD	earth mover distance on sentences		✓	✓	✓			
content overlap	PYRAMID				✓				
	SPICE	scene graph similarity		✓					
	SPIDER	scene graph similarity		✓					

Table 3.1: Untrained *automatic* and *retrieval-based* metrics based on string match, string distance, or context overlap. The acronyms for some of the NLP sub-research fields that each metric is commonly used to evaluate text generation are: **MT**: Machine Translation, **QG**: Question Generation, **SUM**: Summarization, **RG**: Dialog Response Generation, **DG**: Document or Story Generation, Visual-Story Generation, etc., **IC**: Image Captioning.

3.1.1 F-SCORE (F_1)

The F-SCORE, also called the F1-score or F-measure, is a measure of accuracy. The F-SCORE balances the generated text's precision and recall by measuring the harmonic mean of the two measures. F-SCORE is defined as:

$$F_1 = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.1}$$

Precision (specificity), also called the positive predictive value, is the fraction of *n*-grams in the model-generated (hypothesis) text that are present in the reference (human or gold) text. Recall, also called sensitivity, is the fraction of the *n*-grams in the reference text that are present in the candidate text. The F-SCORE reaches the best value, indicating perfect precision and recall, at a value of 1. The worst F-SCORE, which means lowest precision and lowest recall, would be a value of 0.

In text generation tasks such as machine translation or summarization, F-SCORE gives an indication as to the quality of the generated sequence that a model will produce (Melamed et al., 2003; Aliguliyev, 2008). Specifically for machine translation, F-SCORE based metrics have been shown to be effective in evaluating translation quality. One of these metrics is the **CHRF**, character *n*-gram F-score, which uses character *n*-grams instead of word *n*-grams to compare the machine translation model output with the reference translations (Popović, 2015). They use character *n*-grams because it helps to better match the morphological variations in words. In recent work by Mathur et al. (2020), it was empirically shown that CHRF has high correlation with human judgments compared to commonly used *n*-gram-based evaluation metrics.

3.1.2 BLEU

The Bilingual Evaluation Understudy (BLEU) is one of the first metrics used to measure the similarity between two sentences (Papineni et al., 2002). Originally proposed for machine translation, it compares a candidate translation of text to one or more reference translations. BLEU is a weighted geometric mean of *n*-gram precision scores, defined as:

$$prec_n = \frac{\sum_s min(c(s,\hat{y}), c(s,y))}{\sum_s c(s,\hat{y})} \tag{3.2}$$

where $\hat{y}$ is the hypothesis sequence, y is the ground-truth sequence, s is an n-gram sequence of $\hat{y}$, and $c(s, \hat{y})$ is the number of times s appears in $\hat{y}$. The BLEU score is then:

$$\text{BLEU} = \mathcal{BP} \cdot exp(\sum_{n=1}^{N} w_n \log prec_n) \tag{3.3}$$

$\mathcal{BP}$ is the *brevity* penalty to penalize sequences that are too short, and often calculated as:

$$\mathcal{BP} = \begin{cases} 1, & \hat{T} > T \\ e^{(1-r/c)}, & \hat{T} \leq T \end{cases} \tag{3.4}$$

where $\hat{T}$ and T are the prediction and gold sequence length, c is the length of the candidate generated sequence, and r is the effective reference corpus length. N is the total number of n-gram precision scores to use, and w_n is the weight for each precision score, which is often set to be $1/N$.

It has been argued that although BLEU has significant advantages, it may not be the ultimate measure for improved machine translation quality (Callison-burch & Osborne, 2006). Earlier work has reported that BLEU correlates well with human judgments (Lee & Przybocki, 2005; Denoual & Lepage, 2005). More recent work argues that although it can be a good metric for the machine translation task (Zhang et al., 2004) for which it is designed, it doesn't correlate well with human judgments for other generation tasks outside of machine translation (such as image captioning or dialog response generation). Reiter (2018) report that there's not good evidence to support that BLEU is the best metric for evaluating NLG systems other than machine translation. In Caccia et al. (2018), it was empirically demonstrated that, when used to sample from the model outputs that has perfect BLEU with the corpus, the generated sentences were grammatically correct, but lacked semantic and/or global coherence, concluding that the generated text has poor information content.

Outside of machine translation, BLEU has been used for other text generation tasks, such as document summarization (Graham, 2015), image captioning (Vinyals et al., 2014), human-machine conversation (Gao et al., 2019), and language generation (Semeniuta et al., 2019). In Graham (2015), it was concluded that BLEU achieves strongest correlation with human assessment, but does not significantly outperform the best-performing ROUGE variant. On the other hand, a more recent study has demonstrated that n-gram matching scores such as BLEU can be insufficient and potentially less accurate metric for unsupervised language generation (Semeniuta et al., 2019).

Text generation research, especially when focused on short text generation like sentence-based machine translation or question generation, has successfully used BLEU for benchmark analysis with models since it is fast, easy to calculate, and enables a comparison with other models on the same task. However, BLEU has some drawbacks for NLG tasks where contextual understanding and reasoning is the key (e.g., story generation (Fan et al., 2018; Martin et al., 2017) or long-form question answering (Fan et al., 2019a)). It considers neither semantic meaning nor sentence structure. It does not handle morphologically rich languages well, nor does it map well to human judgments (Tatman, 2019). Recent work by (Mathur et al., 2020) investigated how sensitive the machine translation evaluation metrics are to outliers. They found that when there are outliers in tasks like machine translation, metrics like BLEU lead to high correlations yielding false conclusions about reliability of these metrics. They report that when the outliers are removed, these metrics do not correlate as well as before, which adds evidence to the unreliablity of BLEU. We will present other metrics that address some of these shortcomings throughout this paper.

3.1.3 NIST

Proposed by the US National Institute of Standards and Technology, NIST (Martin & Przybocki, 2000) is a method similar to BLEU for evaluating the quality of text. Unlike BLEU, which treats each n-gram equally, NIST heavily weights n-grams that occur less frequently, as co-occurrences of these n-grams are more informative than common n-grams (Doddington, 2002). Information weights are computed using n-gram counts over the set of reference translations, according to the following equation:

$$\text{Info}(w_1 \cdots w_n) = \log_2 \frac{c(w_1 \cdots w_{k-1})}{c(w_1 \cdots w_n)} \tag{3.5}$$

where w_i are *n*-grams in the reference text and $c(\cdot)$ indicates count. The formula for calculating the NIST score is:

$$\text{NIST} = \sum_{n=1}^{N} \left[\frac{\sum_{\substack{\text{all} w_1 \ldots w_s \\ \text{that co-occur}}} \text{Info}(w_1 \cdots w_n)}{\sum_{\substack{\text{all} w_1 \cdots w_n \\ \text{in hyp. sequence}}} (1)} \right] \cdot \exp \left\{ \mathcal{BR} \cdot \log \left[\min \left(\frac{L_{\text{hyp}}}{L_{\text{ref}}}, 1 \right) \right] \right\} \tag{3.6}$$

where L_{hyp} is the average number of words in a hypothesis (generated) translation, averaged over all reference translations, and L_{ref} is the number of words in the translation being scored.

Different from BLEU, the brevity penalty is chosen to be 0.5 when the number of words in the reference output is two-thirds of the average number of words in the reference translation. This change is made to minimize the impact on the score when there is a small variation in the translation length. The goal is to preserve the original motivation of using the brevity penalty, while reducing the contributions of length variations on the score when there are small variations. It has shown that the NIST metric is often superior to BLEU in terms of reliability and quality (Doddington, 2002). Even though this metric has several merits in evaluating machine translation, it has not been adopted by recent neural NLG research as much as the BLEU metric.

3.1.4 ROUGE

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004) is a set of metrics for evaluating automatic summarization of long texts consisting of multiple sentences or paragraphs. Although mainly designed for evaluating single- or multi-document summarization, it has also been used for evaluating short text generation, such as machine translation (Lin & Och, 2004), image captioning (Cui et al., 2018), and question generation (Nema & Khapra, 2018; Dong et al., 2019).

ROUGE includes a large number of distinct variants, including eight different *n*-gram counting methods to measure *n*-gram overlap between the generated and the ground-truth (human-written) text. Simplifying the notation in the original paper (Lin, 2004), ROUGE-N can be defined as:

$$\text{ROUGE-N} = \frac{\sum_r \sum_n \text{match}\left(gram_{n,r}\right)}{\sum_r \sum_s \text{count}\left(gram_n\right)} \tag{3.7}$$

where $\sum_n$ sums over all *n*-grams of length n (e.g., if $n = 2$, the formula measures the number of times a matching *bigram* is found in the hypothesis (model-generated) and the reference (human-generated) text). If there is more than one reference summary, the outer summation ($\sum_r$) repeats the process over all reference summaries. We explain commonly used ROUGE metrics (Lin, 2004) below:

- **ROUGE-1** measures the overlap of *unigrams* (single tokens) between the reference and hypothesis text (e.g,. summaries).
- **ROUGE-2** measures the overlap of *bigrams* between the reference and hypothesis text.
- **ROUGE-3** measures the overlap of *trigrams* between the reference and hypothesis text.
- **ROUGE-4** measures the overlap of *four-grams* between the reference and hypothesis text.
- **ROUGE-L** measures the longest matching sequence of words using longest common subsequence (LCS). This metric doesn't require consecutive matches, but it requires in-sequence matches that indicate sentence-level word order. The *n*-gram length does not need to be predefined since ROUGE-L automatically includes the longest common *n*-grams shared by the reference and hypothesis text.
- **ROUGE-W** (less commonly used) measures weighted LCS-based statistics that favor consecutive LCSs.
- **ROUGE-S** (less commonly used) measures skip-bigram[1]-based co-occurrence statistics. Any pair of skip-words in the sentence order is considered a skip-bigram.

[1] A *skip-gram* (Huang et al., 1992) is a type of *n*-gram in which tokens (e.g., words) don't need to be consecutive but in order in the sentence, where there can be gaps between the tokens that are skipped over. In NLP research, they are used to overcome data sparsity issues.

- **ROUGE-SU** (less commonly used) measures skip-bigram and unigram-based co-occurrence statistics.

ROUGE also includes a setting for word-stemming of summaries and an option to remove or retain stop-words. Additional configurations include the use of precision (ROUGE-P), recall (ROUGE-R), or F-score (ROUGE-F) to compute individual summary scores. It also provides options for computation of the overall score for a system by computing the mean or median of the generated (hypothesis) text score distribution, which is not found in BLEU scores. In total, ROUGE can provide 192 (8 x 2 x 2 x 3 x 2) possible system-level measure variants.

Compared to BLEU, ROUGE focuses on recall rather than precision and is more interpretable than BLEU (Callison-burch & Osborne, 2006). Additionally, ROUGE includes the mean or median score from individual output text, which allows for a significance test of differences in system-level ROUGE scores, while this is restricted in BLEU (Graham & Baldwin, 2014; Graham, 2015). ROUGE evaluates the adequacy of the generated output text by counting how many *n*-grams in the generated output text matches the *n*-grams in the reference (human-generated) output text. This is considered a bottleneck of this measure, especially for long-text generation tasks (Kilickaya et al., 2017), because it doesn't provide information about the narrative flow, grammar, or topical flow of the generated text, nor does it evaluate the factual correctness of the summary compared to the corpus it is generated from.

3.1.5 METEOR

The Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Lavie et al., 2004; Banerjee & Lavie, 2005) is a metric designed to address some of the issues found in BLEU and has been widely used for evaluating machine translation models and other models, such as image captioning (Kilickaya et al., 2017), question generation (Nema & Khapra, 2018; Du et al., 2017), and summarization (See et al., 2017; Chen & Bansal, 2018; Yan et al., 2020). Compared to BLEU, which only measures the precision, METEOR is based on the harmonic mean of the unigram precision and recall, in which recall is weighted higher than precision. Several metrics support this property since it yields high correlation with human judgments (Lavie & Agarwal, 2007).

The METEOR score between a reference and hypothesis text is measured as follows. Let $c(u)$ represent the unigrams found between the hypothesis and reference text, $c_h(u)$ be the total number of unigrams in the hypothesis text, and $c_r(u)$ be the total number of unigrams in the reference text (r). The mean F-SCORE is computed using the unigram Recall=$c(u)/c_r(u)$ and Precision=$c(u)/c_h(u)$:

$$\mathrm{F}_{mean} = \frac{\text{Precision} * \text{Recall}}{\alpha \cdot \text{Precision} + (1-\alpha) \cdot \text{Recall}} \tag{3.8}$$

Then, the alignment between the hypothesis and reference is calculated as follows:

$$\text{METEOR} = \mathrm{F}_{mean} \cdot (1 - penalty) \tag{3.9}$$

The penalty term, which is called the fragmentation penalty, determines the extent to which the matched unigrams in both hypothesis and reference are well-ordered and is measured as follows:

$$penalty = \gamma \cdot frag^{\beta} \tag{3.10}$$

In the above, first, the sequence of matched unigrams between the two texts is divided into the fewest possible number of chunks, such that the matched unigrams in each chunk are adjacent and identical in word order. The number of chunks (ch) and the number of matches (m) are then used to calculate a fragmentation fraction: $frag = ch/m$. γ determines the maximum penalty ($0 \leq \gamma \leq 1$), and β determines the functional relation between the fragmentation and the penalty. METEOR scores range between 0 and 1.

METEOR has several variants that extend exact word matching that most of the metrics in this category do not include, such as stemming and synonym matching. These variants address the problem of reference translation variability, allowing for morphological variants and synonyms to be recognized as valid translations. The metric has been found to produce good correlation with human judgments at the sentence or segment level (Agarwal & Lavie, 2008). This differs from BLEU in that METEOR is explicitly designed to compare at the sentence level rather than the corpus level.

3.1.6 HLEPOR

Harmonic mean of enhanced Length Penalty, Precision, *n*-gram Position difference Penalty, and Recall (HLEPOR), initially proposed for machine translation, is a metric designed for morphologically complex languages like Turkish or Czech (Han et al., 2013b). Among other factors, HLEPOR utilizes part-of-speech (noun, verb, etc.) tags' similarity to capture syntactic information.

3.1.7 RIBES

Rank-based Intuitive Bilingual Evaluation Score (RIBES) (Isozaki et al., 2010) is another un-trained automatic evaluation metric for machine translation. It was developed by NTT Communication Science Labs and designed to be more informative for Asian languages——like Japanese and Chinese—since it doesn't rely on word boundaries. Specifically, RIBES is based on how the words in generated text are ordered. It uses the rank correlation coefficients measured based on the word order from the hypothesis (model-generated) translation and the reference translation. Some of the correlation coefficients used in RIBES are Spearman's ρ, which is based on the distance of difference in the ranks, or Kendall's τ, which is based on the direction of the difference in rank. However, earlier work on evaluating the correlation of automatic metrics with human judgments has shown that RIBES tends to show lower correlation with human evaluation scores, indicating that higher RIBES doesn't necessary yield better translations (Tan et al., 2015).

3.1.8 CIDEr

Consensus-based Image Description Evaluation (CIDEr) is an automatic metric for measuring the similarity of a generated sentence against a set of human-written sentences using a consensus-based protocol. Originally proposed for image captioning (Vedantam et al., 2014), CIDEr shows high agreement with consensus as assessed by humans. It enables a comparison of text generation models based on their "human-likeness," without having to create arbitrary calls on weighing content, grammar, saliency, etc. with respect to each other.

The CIDEr metric presents three explanations about what a hypothesis sentence should contain:

1. *n*-grams in the hypothesis sentence should also occur in the reference sentences.
2. If an *n*-gram does not occur in a reference sentence, it shouldn't be in the hypothesis sentence.
3. *n*-grams that commonly occur across all image-caption pairs in the dataset should be assigned lower weights, since they are potentially less informative.

Given these intuitions, a Term Frequency Inverse Document Frequency (TF-IDF) weight is calculated for each *n*-gram. Specifically, given an image i and a list of the reference descriptive sentences about the image, $s_{ij} \in S_i = \{s_{i1}, ..., s_{im}\}$, where $h_k(s_{ij})$ and $h_k(c_i)$ represent the numbers of times an *n*-gram w_k occurs in a reference sentence s_{ij} and a hypothesis (model-generated) sentence c_i, respectively. The TF-IDF score is calculated as follows:

$$g_k(s_{ij}) = \frac{h_s(s_{ij})}{\sum_{w_l \in \omega} h_l(s_{ij})} \log \left(\frac{|I|}{\sum_{I_p \in I} \min(1, \sum_q h_k(s_{pq}))} \right) \tag{3.11}$$

where I is the set of all images, and ω the vocabulary of all *n*-grams. Then, the CIDEr$_n$ score for a particular *n*-gram is calculated as the cosine similarity between the generated candidate sentence and the reference sentence, as

$$\text{CIDEr}_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{\mathbf{g}^n(c_i) \cdot \mathbf{g}^n(s_{ij})}{||\mathbf{g}^n(c_i)||||\mathbf{g}^n(s_{ij})||} \tag{3.12}$$

where $\mathbf{g}^n(c_i)$ is a vector of $g_k(c_i)$, containing all n-grams of length n, and $||\mathbf{g}^n(c_i)||$ measures the magnitude of the vector $\mathbf{g}^n(c_i)$. To capture richer semantics and grammatical properties, CIDEr can also use higher-order (longer) n-grams as

$$\text{CIDEr}_n(c_i, S_i) = \sum_{n=1}^{N} w_n \text{CIDE}r_n(c_i, S_i) \tag{3.13}$$

Vedantam et al. (2014) find that uniform weights $w_n = 1/N$ work the best where N=4.

A recent study (Kilickaya et al., 2017) shows that among all the untrained automatic metrics for image captioning evaluation, CIDEr is the most robust and correlates the best with human judgments.

3.2 Distance-Based Evaluation Metrics for Content Selection

A distance-based metric in NLG applications uses a distance function to measure the similarity between two text units (e.g., words, sentences). First, we represent two text units using vectors. Then, we compute the distance between the vectors. The smaller the distance, the more similar the two text units are. This section reviews distance-based similarity measures where text vectors can be constructed using discrete tokens, such as bag of words (§3.2.1) or embedding vectors (§3.2.2). We note that even though the embeddings that are used by these metrics to represent the text vectors are pre-trained, these metrics are not trained to mimic the human judgments as in machine-learned metrics that we summarize in Chapter 4.

3.2.1 Edit Distance-Based Metrics

Edit distance, one of the most commonly used evaluation metrics in natural language processing, measures how dissimilar two text units are based on the minimum number of operations required to transform one text into the other. We summarize some of the well-known edit distance measures below.

WER Word error rate (WER), originally designed for measuring the performance of speech recognition systems, is also commonly used to evaluate the quality of machine translation systems (Tomás et al., 2003). Specifically, WER is the percentage of words that need to be inserted, deleted, or replaced in the translated sentence to obtain the reference sentence, i.e., the edit distance between the reference and hypothesis sentences. It is calculated as:

$$\text{WER} = \frac{\text{substitutions + insertions + deletions}}{\text{reference sentence length}} \tag{3.14}$$

where a substitution replaces one word with another, an insertion adds one new word, and a deletion drops one word. The main drawback of WER is its dependency on the reference sentences. In machine translation, there may exist multiple correct translations for the same input. But this metric only considers one to be correct.

Other variations of WER, such as Sentence Error Rate (SER), measure the percentage of sentences whose translations do not exactly match the reference sequence. Multi-reference word error rate (mWER) (Ali et al., 2015) calculates the edit distance between several references for each sentence and chooses the smallest one (Nießen et al., 2000). One drawback of this approach is that it requires human effort to obtain multiple references, but it has been found to be an effective measure. All-reference word error rate (aWER) (Tomás et al., 2003) measures the number of words to be inserted, deleted, or replaced in the sentence under evaluation in order to obtain a correct translation. aWER can be considered a version of mWER which takes into account all possible references, not just one reference, as in WER.

WER has some limitations. For instance, while its value is lower-bounded by zero, which indicates a perfect match between the hypothesis and reference text, its value is not upper-bounded, making it hard to evaluate in an absolute manner (Mccowan et al., 2004). It is also reported to suffer from weak correlation with human evaluation. For example, in the task of spoken document retrieval, the

WER of an automatic speech recognition system is reported to poorly correlate with the retrieval system performance (Kafle & Huenerfauth, 2017).

MED The minimum edit distance (MED) between two text strings is the minimum number of editing operations (i.e., insertion, deletion, and substitutions) required to transform one string into the other. For two strings x and y of length n and m, respectively, we define a distance metric, $D(i, j)$, which will be the edit distance between $x[1 \cdots i]$ (i.e., the first i characters of string x) and $y[1 \cdots j]$ (i.e., the first j characters of string y). Now the distance between the entire two strings x and y will be $D(n, m)$. The MED can be applied to sentences or longer text using words as units rather than characters. In machine translation, MED is the minimum number of insertions, deletions, and substitutions of words that are required in order to make a system translation equivalent in meaning to that of a reference translation. Both WER and MED are based on Levenshtein Distance. While MED is used mainly to measure two text strings, WER is for both text and speech.

TER Translation edit rate (TER) (Snover et al., 2006) is defined as the minimum number of edits needed to change a generated text so that it exactly matches one of the references, normalized by the average length of the references. In terms of minimum number of edits, TER measures the number of edits to the closest reference as:

$$\text{TER} = \frac{\text{number of edits}}{\text{average number of reference words}} \tag{3.15}$$

TER considers the insertion, deletion, and substitution of single words and shifts of words as possible edits. The word shifting moves a contiguous sequence of words within the hypothesis to another location within the hypothesis. The metric assigns all edits an equal cost.

While TER has been shown to correlate well with human judgments in evaluating machine translation quality, it suffers from some limitations. For example, it can only capture similarity in a narrow sense, as it only uses a single reference translation and considers only exact word matches between the hypothesis and the reference. This issue can be partly addressed by constructing a lattice of reference translations, a technique that has been used to combine the output of multiple translation systems (Rosti et al., 2007). Many variants have been proposed to improve the original TER. TERP[2] uses phrasal substitutions (using automatically generated paraphrases), stemming, synonyms, relaxed shifting constraints, and other improvements. ITER (Panja & Naskar, 2018) adds stem matching and normalization on top of TER. CDER (Leusch et al., 2006) models block ordering as an edit operation. PER (Tillmann et al., 1997) computes position independent error rate. Two recent variants, CharacTER (Wang et al., 2016), a character-based translation edit distance measure, and EED (Stanchev et al., 2019), an extension of Levenshtein distance, have shown to correlate better with human judgments on some languages.

3.2.2 Vector Similarity-Based Evaluation Metrics

In NLP, embedding-based similarity measures are commonly used in addition to *n*-gram-based similarity metrics. Embeddings are real-valued vector representations of character or lexical units, such as word-tokens or *n*-grams, that allow tokens with similar meanings to have similar representations. Even though the embedding vectors are learned using supervised or unsupervised neural network models, the vector-similarity metrics we summarize below assume the embeddings are pre-trained and simply used as input to calculate the metric.

MEANT 2.0 The vector-based similarity measure MEANT uses word embeddings and shallow semantic parses to compute lexical and structural similarity (Lo, 2017). It evaluates translation adequacy by measuring the similarity of the semantic frames and their role fillers between the human references and the machine translations.

YISI Inspired by the MEANT score, YISI[3] (Lo, 2019) is proposed to evaluate the accuracy of machine translation model outputs. It is based on the weighted distributional lexical semantic similarity, as well as shallow semantic structures. Specifically, it extracts the longest common character substring from the hypothesis and reference translations to measure the lexical similarity.

[2] TERP is named after the University of Maryland mascot, the Terrapin.

[3] YiSi, is the romanization of the Cantonese word 意思, which translates as ‘meaning’ in English.

Word Mover's Distance (WMD) Earth mover's distance (EMD), also known as the Wasserstein metric (Rubner et al., 1998), is a measure of the distance between two probability distributions. Word mover's distance (WMD; Kusner et al., 2015) is a discrete version of EMD that calculates the distance between two sequences (e.g., sentences, paragraphs, etc.), each represented with relative word frequencies. It combines item similarity[4] on bag-of-word (BOW) histogram representations of text (Goldberg et al., 2018) with word embedding similarity. In short, WMD has several intriguing properties:

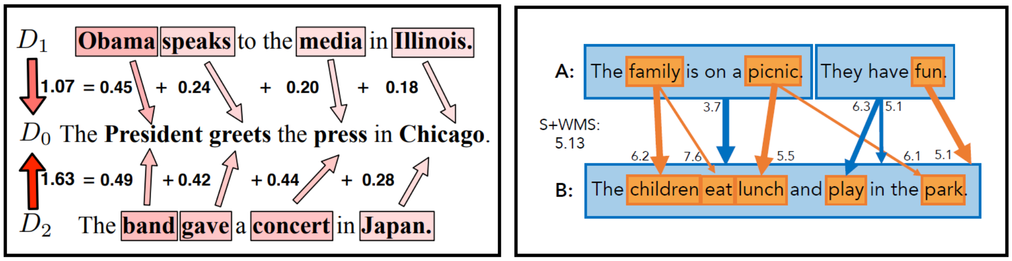
- It is hyperparameter-free and easy to use.
- It is highly interpretable as the distance between two documents can be broken down and explained as the sparse distances between few individual words.
- It uses the knowledge encoded within the word embedding space, which leads to high retrieval accuracy.

For any two documents A and B, we define the WMD as the minimum cost of transforming one document into the other. Each document is represented by the relative frequencies of the words it contains, i.e., for the ith word type,

$$d_{A,i} = \text{count}(i)/|A| \tag{3.16}$$

In Equation 3.16, $|A|$ is the total word count of document A, and $d_{B,i}$ is defined in the same way.

Representing the ith word by $\mathbf{v}_i \in \mathbb{R}^m$, i.e., an m-length embedding,[5] we define distances between the ith and jth words as $\Delta(i,j)$. V is the vocabulary size. Kusner et al. (2015) use the Euclidean distance ($\Delta(i,j) = \|\mathbf{v}_i - \mathbf{v}_j\|_2$).



D_1 Obama speaks to the media in Illinois.
1.07 = 0.45 + 0.24 + 0.20 + 0.18
D_0 The President greets the press in Chicago.
1.63 = 0.49 + 0.42 + 0.44 + 0.28
D_2 The band gave a concert in Japan.
A: The family is on a picnic. They have fun.
S+WMS: 5.13
3.7
6.3
5.1
6.2
7.6
5.5
6.1
5.1
B: The children eat lunch and play in the park.

Figure 3.1: (LEFT) Illustration of Word Mover's Distance (WMD). Picture source: (Kusner et al., 2015); (RIGHT) Illustration of Sentence Mover's Distance (SMD). Picture source: (Clark et al., 2019).

The WMD is then calculated by finding the solution to the linear program:

$$\textsc{wmd}(A,B) = \min_{\mathbf{T}\geq\mathbf{0}} \textstyle\sum_{i=1}^{V} \sum_{j=1}^{V} \mathbf{T}_{i,j}\Delta(i,j) \tag{3.17a}$$

s.t.

$$\forall i, \textstyle\sum_{j=1}^{V} \mathbf{T}_{i,j} = d_{A,i}, \tag{3.17b}$$

$$\forall j, \textstyle\sum_{i=1}^{V} \mathbf{T}_{i,j} = d_{B,j} \tag{3.17c}$$

$\mathbf{T} \in \mathbb{R}^{V\times V}$ is a non-negative matrix, where each $\vec{T}_{i,j}$ denotes how much of word i (across all its tokens) in A is assigned to word j in B, and the constraints ensure the flow of a given word cannot

[4] The similarity can be defined as cosine, Jaccard, Euclidean, etc.

[5] One could use pre-trained type-based or contextual word embeddings.

exceed its weight. Specifically, WMD ensures that the entire outgoing flow from word i equals $d_{A,i}$, i.e., $\sum_j \vec{T}_{i,j} = d_{A,i}$. Additionally, the amount of incoming flow to word j must match $d_{B,j}$, i.e., $\sum_i \vec{T}_{i,j} = d_{B,j}$.

Empirically, WMD has been instrumental to the improvement of many NLG tasks, specifically sentence-level tasks, such as image caption generation (Kilickaya et al., 2017) and natural language inference (Sulea, 2017). However, while WMD works well for short texts, its cost grows prohibitively as the length of the documents increases, and the BOW approach can be problematic when documents become large as the relation between sentences is lost. By only measuring word distances, the metric cannot capture information conveyed in the group of words, for which we need higher-level document representations (Dai et al., 2015; Wu et al., 2018).

Sentence Mover's Distance (SMD) Sentence Mover's Distance (SMD) is an automatic metric based on WMD to evaluate text in a continuous space using sentence embeddings (Clark et al., 2019; Zhao et al., 2019). SMD has been used to compare the generated texts to reference texts in tasks like machine translation and summarization, and is found to be correlated with human evaluation. SMD represents each document as a collection of sentences or of both words and sentences (as seen in Figure 3.1), where each sentence embedding is weighted according to its length. The bag of words and sentences representing document A is normalized by $2|A|$, so that:

$$d_{A,i} = \begin{cases} count(i)/2|A|, & \text{if } i \text{ is a word} \\ |i|/2|A|, & \text{if } i \text{ is a sentence} \end{cases} \tag{3.18}$$

Like WMD, SMD also tries to solve the same linear program in Eq. 3.17. Unlike WMD, SMD measures the cumulative distance of moving both the words in a document and the sentences to match another document. The vocabulary is defined as a set of sentences and words in the documents. On a summarization task, SMD is found to correlate better with human judgments than ROUGE (Clark et al., 2019).

Recently, Zhao et al. (2019) propose a new version of SMD that attains higher correlation with human judgments. Similar to SMD, they use word and sentence embeddings by taking the average of the token-based embeddings before the mover's distance is calculated. They also investigate different contextual embeddings models including ELMO and BERT by taking power mean (which is an embedding aggregation method) of their embeddings at each layer of the encoding model.

Fréchet Inception Distance Extending Inception score, Heusel et al. (2017) propose a new metric called Fréchet Inception Distance (FID) to score the similarity between generated images and real ones. It measures the distance between two multivariate Gaussians:

$$FID = ||\mu_r - \mu_g||^2 + Tr\left(\sum_r + \sum_g - 2(\sum_r \sum_g)\right)^{1/2} \tag{3.19}$$

where the samples $X_r \sim \mathcal{N}(\mu_r, \sum_r)$ and $X_g \sim \mathcal{N}(\mu_g, \sum_g)$ are the hidden-layer activations of the Inception v3 for real and generated samples, respectively. The authors assume that the features extracted by a classifier are normally distributed. Semeniuta et al. (2018) adapt FID to NLG evaluation by using InferSent text embedding model (Conneau et al., 2017) to compute the sentence embeddings. InferSent is a supervised model of bidirectional LSTM with max pooling.

3.3 *n*-gram-Based Diversity Metrics

The lexical diversity score measures the breadth and variety of the word usage in writing (Inspector, 2013). Consider two pieces of texts about in-class teaching. The first repeatedly uses the same words such as '*teacher*', '*reads*', and '*asks*'. The second one avoids repetition by using different words or expressions, e.g, '*lecturer*', '*instructor*', '*delivers*', '*teaches*', '*questions*', '*explains*', etc. The second text is more lexically diverse, which is more desirable in many NLG tasks such as conversational bots (Li et al., 2018), story generation (Rashkin et al., 2020), question generation (Du et al., 2017; Pan et al., 2019), and abstractive question answering (Fan et al., 2019).

In this section we review some of the metrics designed to measure the quality of the generated text in terms of lexical diversity.

3.3.1 Type-Token Ratio (TTR)

Type-Token Ratio (TTR) is a measure of lexical diversity (Richards, 1987), mostly used in linguistics to determine the richness of a writer's or speaker's vocabulary. It is computed as the number of unique words (types) divided by the total number of words (tokens) in a given segment of language:

$$\text{TTR}(\text{text}) = \frac{\#\text{distinct tokens}}{\#\text{total tokens}} \tag{3.20}$$

Although intuitive and easy to use, TTR has a major problem: it is sensitive to text length. The longer the document, the lower the prospect that a replacement token will be a new type. This eventually causes the TTR to drop as more words are added. To remedy this issue, several other lexical diversity measures have been proposed, and we discuss them below.

Measuring diversity using *n*-gram repetitions is a more generalized version of TTR, which has been use for text generation evaluation. Li et al. (2016) has shown that modeling mutual information between source and targets significantly decreases the chance of generating bland responses and improves the diversity of responses. They use BLEU and distinct word unigram and bigram counts to evaluate the proposed diversity-promoting objective function for dialog response generation.

3.3.2 SELF-BLEU

Zhu et al. (2018) propose SELF-BLEU as a diversity evaluation metric by measuring the differences between generated sentences and references or other generated texts. In a sense, it is the opposite of BLEU, which assesses how similar two sentences are. Taking a generated sentence to be evaluated as the hypothesis and the other sentences as references, SELF-BLEU calculates a BLEU score for every generated sentence and defines the average of these BLEU scores as the SELF-BLEU score of the to-be-evaluated text. A lower SELF-BLEU score implies higher diversity.

Several NLG papers have reported that SELF-BLEU achieves good generation diversity (Zhu et al., 2018; Chen et al., 2018; Lu et al., 2018). However, others have reported some weakness of the metric in generating diverse output (Caccia et al., 2018) or detecting mode collapse (Semeniuta et al., 2019) in text generation with GAN (Goodfellow et al., 2014) models. Even though SELF-BLEU is mainly used for evaluating the diversity of generated sentences, people are exploring a better evaluation metric that evaluates both quality and diversity (Montahaei et al., 2019a).

3.3.3 Measure of Textual Lexical Diversity

As we noted earlier in this chapter, the TTR metric is sensitive to the length of the text. To remedy this, a new diversity metric, HD-D (hyper-geometric distribution function), is proposed to compare texts of different lengths (McCarthy & Jarvis, 2010).

McCarthy & Jarvis (2010) argue that the probabilities of word occurrence can be modeled using the hyper-geometric distribution (HD). The HD is a discrete probability distribution that expresses the probability of k successes after drawing n items from a finite population of size N containing m successes without replacement. HD is used to measure lexical diversity, entitled HD-D. HD-D assumes that if a text sample consists of many tokens of a specific word, then there is a high probability of drawing a text sample that contains at least one token of that word. This measure does not require a minimum k tokens to be estimated.

The HD-D and its variants (McCarthy & Jarvis, 2010) have been used to measure the diversity in story generation (McCarthy & Jarvis, 2010) and summarization tasks (Crossley et al., 2019).

3.4 Explicit Semantic Content Match Metrics

Semantic content matching metrics define the similarity between human-written and model-generated text by extracting explicit semantic information units from text beyond *n*-grams. These metrics operate on semantic and conceptual levels, and are shown to correlate well with human judgments. We summarize some of them below.

3.4.1 PYRAMID

The PYRAMID method is a semi-automatic evaluation method (Nenkova & Passonneau, 2004) for evaluating the performance of document summarization models. Like other untrained automatic metrics that require references, this untrained metric also requires human annotations. It identifies summarization content units (SCUs) to compare information in a human-generated reference summary to the model-generated summary. To create a pyramid, annotators begin with model-generated summaries of the same source texts and select sets of text spans that express the same meaning across summaries. Each set is referred to as a SCU and receives a label for mnemonic purposes. An SCU has a weight corresponding to the number of summaries that express the SCU's meaning.

SCUs are extracted from a corpus of summaries by annotators and are not longer than a clause (see Table 3.2). The annotation starts with identifying similar sentences and then proceeds with more fine-grained inspection that identifies related sub-parts. The SCUs that appear in human-generated summaries more often get higher weights. So a pyramid is formed after the SCU annotation of human-generated summaries. The SCUs that appear in most of the summaries appear at the top of the pyramid and get the greatest weights. The lower in the pyramid an SCU appears, the lower its weight is because it occurs in fewer summaries. The SCUs in peer summary are then checked against an existing pyramid to measure how much information agrees between the model-generated and human-generated summaries.

SCU: The cause of an airline crash over Nova Scotia has not been determined
a. *The cause of the Sept. 2, 1998*$_{A2}$ **crash has not been determined.**
b. *searched for clues as to a cause*$_{A2}$ **but refrained from naming one.**
c. **The cause has not been determined,**
d. **The specific cause of the tragedy was never determined**
e. **but investigators remain unsure of its cause.**
f. *A final determination of the crashes cause is still far off.* $_{A1}$

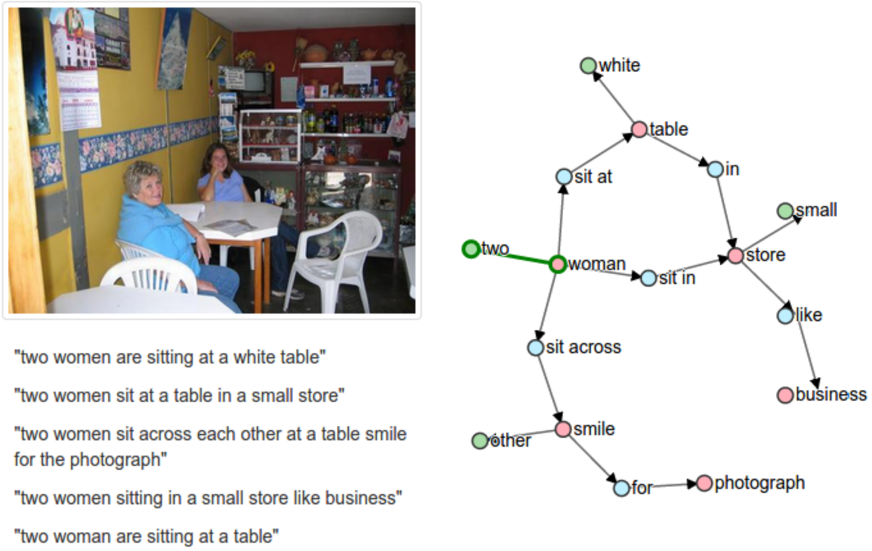
Table 3.2: Overlay of matching SCUs from two annotators $A1$ and $A2$ from summaries a through f. **Boldface** indicates text selected by both annotators. Text spans in *italics* are labeled $A1$ or $A2$ to indicate which annotator selected them. Table Source: (Passonneau, 2006)

The PYRAMID metric relies on manual human labeling effort, which makes it difficult to automate. In a recent study, the PEAK: Pyramid Evaluation via Automated Knowledge Extraction (Yang et al., 2016) is presented as a fully automated variant of PYRAMID model, which can automatically assign the pyramid weights. First, the PEAK identifies relation triples (subject-predicate-object triples) using open information extraction (Del Corro & Gemulla, 2013). Then, these triplets are combined into a hypergraph based on a semantic similarity approach in which the nodes become the triplets. Salient nodes on the graph, which are later assigned as potential SCUs, are determined based on novel similarity metrics defined on the graph. The PEAK metric not only generates a pyramid entirely automatically but also is shown to correlate well with human judgments (Yang et al., 2016).

3.4.2 SPICE

Semantic propositional image caption evaluation (SPICE) (Anderson et al., 2016) is an image captioning metric that measures the similarity between a list of reference human written captions $S = \{s_1, \cdots, s_m\}$ of an image and a hypothesis caption c generated by a model. Instead of directly comparing a generated caption to a set of references in terms of syntactic agreement, SPICE parses each reference to derive an abstract scene graph representation. The generated caption is also parsed and compared to the scene graph to capture the semantic similarity. SPICE has shown to have a strong correlation with human ratings.

A scene graph (Schuster et al., 2015) encodes objects, attributes, and relationships detected in image captions, representing an image in a skeleton form, as shown in Figure 3.2. A two-stage process is typically used to parse an image caption into a scene graph (Schuster et al., 2015; Lin et al., 2014). First, syntactic dependencies between words in the generated and reference captions are extracted using a dependency parser (Klein & Manning, 2003). Second, the extracted dependency tree is


"two women are sitting at a white table"

"two women sit at a table in a small store"

"two women sit across each other at a table smile for the photograph"

"two women sitting in a small store like business"

"two woman are sitting at a table"

Figure 3.2: Illustration of Scene Graph Extraction for measuring the SPICE metric. A scene graph (right) is parsed from a set of reference image captions on the left. Picture source: (Anderson et al., 2016).

mapped to a scene graph using a rule-based system (Schuster et al., 2015). SPICE then computes the F-SCORE using the hypothesis and reference scene graphs over the conjunction of logical tuples representing semantic propositions in the scene graph.

The semantic relations in a scene graph (G) are represented as a conjunction of logical propositions, namely tuples (T), defined as:

$$T(G(c)) = O(c) \cup A(c) \cup R(c) \tag{3.21}$$

Each tuple contains up to three elements, indicating objects (O), relations (R), and attributes (A). A binary matching operator $\otimes$ then returns the matching tuples in two scene graphs. The SPICE metric is defined as:

$$\text{SPICE}(c, S) = \text{F}_1(c, S) = \frac{2 \cdot P(c, S) \cdot R(c, S)}{P(c, S) + R(c, S)} \tag{3.22}$$

where $P(c, S)$ and $R(c, S)$ are the precision and recall.

As discussed in the previous subsections, most evaluation metrics are based on *n*-gram matching, such as BLEU and METEOR. However, a higher number of matched *n*-grams doesn't always indicate a higher generation quality because two sentences with a lot of words in common can be very different semantically. In comparison, SPICE is more desirable because it measures the semantic similarity between a hypothesis and a reference text using scene graphs. Even though SPICE correlates well with human evaluations, a major drawback is that it ignores the fluency of the generated captions (Sharif et al., 2018).

3.4.3 SPIDER

Liu et al. (2017) propose SPIDER which is a linear combination of SPICE and CIDER. They show that optimizing SPICE alone often results in captions that are wordy and repetitive Although scene graph similarity is good at measuring the semantic similarity between captions, it does not take into account the syntactical aspects of texts. Thus, a combination of semantic graph similarity (like SPICE) and *n*-gram similarity measure (like CIDER) yields a more complete quality evaluation metric. However, the correlation of SPIDER and human evaluation is not reported.

3.4.4 Semantic Similarity Models used as Evaluation Metrics

Other text generation work has used the confidence scores obtained from semantic similarity methods as an evaluation metric. Such models can evaluate a reference and a hypothesis text based on their task-level semantics. The most commonly used methods based on the sentence-level similarity are as follows:

- **Semantic Textual Similarity** (STS) is concerned with the degree of equivalence in the underlying semantics of paired text (Agirre et al., 2016). The STS is used as an evaluation metric in text generation tasks such as machine translation, summarization, and dialogue response generation in conversational systems. The official score is based on weighted Pearson correlation between predicted similarity and human-annotated similarity. The higher the score, the better the the similarity prediction result from the algorithm (Maharjan et al., 2017; Cer et al., 2017).
- **Paraphrase identification** (PI) considers if two sentences express the same meaning (Dolan & Brockett, 2005; Barzilay & Lee, 2003). PI is used as a text generation evaluation score based on the textual similarity (Kauchak & Barzilay, 2006) of reference and hypothesis by finding a paraphrase of the reference sentence that is closer in wording to the hypothesis output. For instance, given the pair of sentences:

  reference: ”*However, Israel’s reply failed to completely clear the U.S. suspicions.*”
  hypothesis: ”*However, Israeli answer unable to fully remove the doubts.*”

  PI is concerned with learning to transform the reference sentence into:

  paraphrase: ”*However, Israel’s answer failed to completely remove the U.S. suspicions.*”

  which is closer in wording to the hypothesis. In Jiang et al. (2019), a new paraphrasing evaluation metric, TIGER, is used for image caption generation evaluation. Similarly, considering image captioning, Liu et al. (2019a) introduce different strategies to select useful visual paraphrase pairs for training by designing a variety of scoring functions.
- **Textual entailment** (TE) is concerned with whether a hypothesis can be inferred from a premise, requiring understanding of the semantic similarity between the hypothesis and the premise (Dagan et al., 2006; Bowman et al., 2015). It has been used to evaluate several text generation tasks, including machine translation (Padó et al., 2009), document summarization (Long et al., 2018), language modeling (Liu et al., 2019b), and video captioning (Pasunuru & Bansal, 2017).
- **Machine Comprehension** (MC) is concerned with the sentence matching between a passage and a question, pointing out the text region that contains the answer (Rajpurkar et al., 2016). MC has been used for tasks like improving question generation (Yuan et al., 2017; Du et al., 2017) and document summarization (Hermann et al., 2015).

3.5 Syntactic Similarity-Based Metrics

A syntactic similarity metric captures the similarity between a reference and a hypothesis text at a structural level to capture the overall grammatical or sentence structure similarity.

In corpus linguistics, part of speech (POS) tagging is the process of assigning a part-of-speech tag (e.g., verb, noun, adjective, adverb, and preposition, etc.) to each word in a sentence, based on its context, morphological behaviour, and syntax. POS tags have been commonly used in machine translation evaluation to evaluate the quality of the generated translations. While TESLA (Dahlmeier et al., 2011) is introduced as an evaluation metric to combine the synonyms of bilingual phrase tables and POS tags, others use POS *n*-grams together with a combination of morphemes and lexicon probabilities to compare the target and source translations (Popovic et al., 2011; Han et al., 2013a). POS tag information has been used for other text generation tasks such as story generation (Agirrezabal et al., 2013), summarization (Suneetha & Fatima, 2011), and question generation (Zerr, 2014).

Syntactic analysis studies the arrangement of words and phrases in well-formed sentences. For example, a dependency parser extracts a dependency tree of a sentence to represent its grammatical structure. Several text generation tasks have enriched their evaluation criteria by leveraging syntactic analysis. In machine translation, Liu & Gildea (2005) use constituent labels and head-modifier dependencies to extract structural information from sentences for evaluation, while others use shallow parsers (Lo et al., 2012) or a dependency parser (Yu et al., 2014, 2015). Yoshida et al. (2014) combine a sequential decoder with a tree-based decoder in a neural architecture for abstractive text summarization.

Chapter 4

Machine-Learned Evaluation Metrics

Many of the untrained evaluation metrics described in Chapter 3 assume that the generated text has significant word (or *n*-gram) overlap with the ground-truth text. However, this assumption does not hold for many NLG tasks, such as a social chatbot, which permit significant diversity and allow multiple plausible outputs for a given input. Table 4.1 shows two examples from the dialog response generation and image captioning tasks, respectively. In both tasks, the model-generated outputs are plausible given the input, but they do not share any words with the ground-truth output.

One solution to this problem is to use embedding-based metrics, which measure semantic similarity rather than word overlap, as in Section 3.2.2. But embedding-based methods cannot help in situations when the generated output is semantically different from the reference, as in the dialog example. In these cases, we can build machine-learned models (trained on human judgment data) to mimic human judges to measure many quality metrics of output, such as factual correctness, naturalness, fluency, coherence, etc. In this chapter we survey the NLG evaluation metrics that are computed using machine-learned models, with a focus on recent neural models.

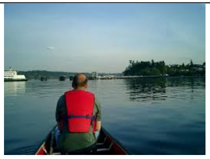
	Dialog Response Generation	Image Captioning
Context	**Speaker A**: Hey John, what do you want to do tonight? **Speaker B**: Why don't we go see a movie?	
Ground-Truth	**Response:** Nah, I hate that stuff, let's do something active.	**Caption:** a man wearing a red life jacket is sitting in a canoe on a lake
Model/Distorted Output	**Response:** Oh sure! Heard the film about Turing is out!	**Caption:** a guy wearing a life vest is in a small boat on a lake
BLEU	0.0	0.20
ROUGE	0.0	0.57
WMD	0.0	0.10

Table 4.1: Demonstration of issues with using automatic evaluation metrics that rely on *n*-gram overlap using two short-text generation tasks: dialog response generation and image captioning. The examples are adapted from Liu et al. (2016) and Kilickaya et al. (2017).

4.1 Sentence Semantic Similarity Based Evaluation

Neural approaches to sentence representation learning seek to capturing semantic and syntactic meanings of sentences from different perspectives and topics and to map a sentence onto an embedding vector using DNN models. As with word embeddings, NLG models can be evaluated by embedding each sentence in the generated and reference texts.

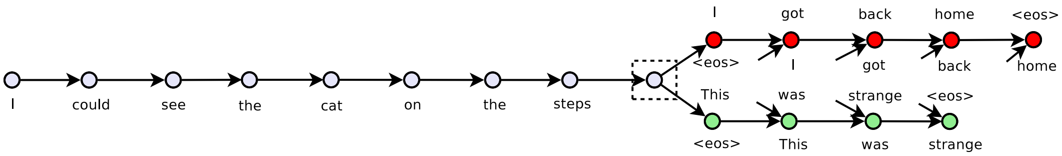


Figure 4.1: Illustration of Skip-Thoughts Vectors Model for sentence representation learning (Image Source: (Kiros et al., 2015)).

Extending word2vec (Mikolov et al., 2013) to produce word or phrase embeddings, one of the earliest sentence embeddings models, Deep Semantic Similarity Model (**DSSM**) (Huang et al., 2013) introduces a series of latent semantic models with a deep structure that projects two or more text streams (such as a query and multiple documents) into a common low-dimensional space where the relevance of one text towards the other text can be computed via vector distance. The **SKIP-THOUGHT** vectors model (Kiros et al., 2015) exploits the encoder-decoder architecture to predict context sentences in an unsupervised manner. Skip-thought vectors allow us to encode rich contextual information by taking into account the surrounding context, but are slow to train. **FASTSENT** (Hill et al., 2016) makes training efficient by representing a sentence as the sum of its word embeddings, but also dropping any knowledge of word order. A simpler **WEIGHTED SUM** of word vectors (Arora et al., 2019) weighs each word vector by a factor similar to the tf-idf score, where more frequent terms are weighted less. Similar to FASTSENT, it ignores word order and surrounding sentences. Extending DSSM models, **INFERSENT** (Conneau et al., 2017) is an effective model, which uses LSTM-based Siamese networks, with two additional advantages over the FASTSENT. It encodes word order and is trained on a high-quality sentence inference dataset. On the other hand, **QUICK-THOUGHT** (Logeswaran & Lee, 2018) is based on an unsupervised model of universal sentence embeddings trained on consecutive sentences. Given an input sentence and its context, a classifier is trained to distinguish a context sentence from other contrastive sentences based on their embeddings.

The recent large-scale pre-trained language models (PLMs) such as **ELMO** and **BERT** use contextualized word embeddings to represent sentences. Even though these PLMs outperform the earlier models such as DSSMs, they are more computationally expensive to use for evaluating NLG systems. For example, the Transformer-based BERT model (Devlin et al., 2018) and its extension ROBERTA (Liu et al., 2019c) are designed to learn textual similarities on sentence-pairs using cosine similarities, similar to DSSM. But both are much more computationally expensive than DSSM due to the fact that they use a much deeper NN architecture, and need to fine-tuned for different tasks. To remedy this, Reimers & Gurevych (2019) propose to use **SENTBERT**, which is a fine-tuned BERT on a "general" task to optimize the BERT parameters, so that a cosine similarity between two generated sentence embeddings is strongly related to the semantic similarity of the two sentences. Then the fine-tuned model can be used to evaluate various NLG tasks. A recent study focusing on machine translation task, ESIM also computes sentence representations from BERT embeddings (with no fine-tuning), and later computes the similarity between the translated text and its reference using metrics such as average recall of its reference. (Chen et al., 2017; Mathur et al., 2019).

4.2 Evaluating Factual Correctness

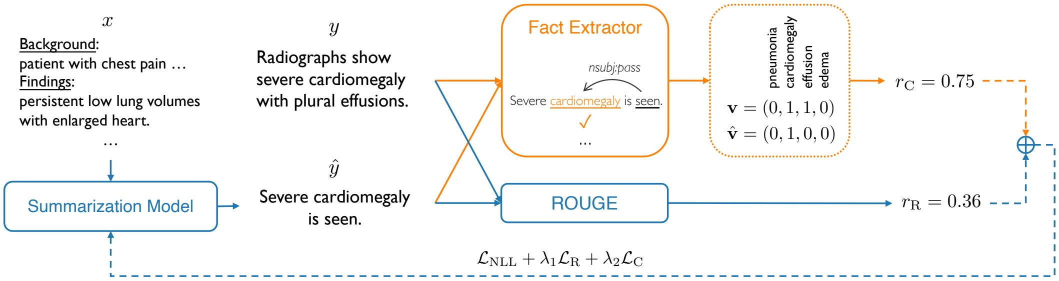


Figure 4.2: Illustration of the training strategy of the factually correct summarization model. Image Source: (Zhang et al., 2019b).

Zhang et al. (2019b) propose a way to tackle the problem of factual correctness in summarization models. Focusing on summarizing radiology reports, they extend pointer networks for abstractive summarization by introducing a reward-based optimization that trains the generators to obtain more rewards when they generate summaries that are factually aligned with the original document. Specifically, they design a fact extractor module so that the factual accuracy of a generated summary can be measured and directly optimized as a reward using policy gradient, as shown in Figure 4.2. This fact extractor is based on an information extraction module and extracts and represents the facts from generated and reference summaries in a structured format. The summarization model is updated via reinforcement learning using a combination of the NLL (negative log likelihood) loss, a ROUGE-based loss, and a factual correctness-based loss ($Loss$=$\mathcal{L}_{NLL}$+$\lambda_1\mathcal{L}_{rouge}$+$\lambda_2\mathcal{L}_{fact}$). Their work suggests that for domains in which generating factually correct text is crucial, a carefully implemented information extraction system can be used to improve the factual correctness of neural summarization models via reinforcement learning.

To evaluate the factual consistency of the text generation models, Eyal et al. (2019b) present a question-answering-based parametric evaluation model named Answering Performance for Evaluation of Summaries (APES). Their evaluation model is designed to evaluate document summarization and is based on the hypothesis that the quality of a generated summary is associated with the number of questions (from a set of relevant ones) that can be answered by reading the summary.

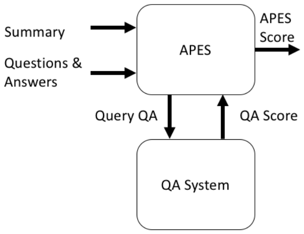


Figure 4.3: APES evaluation flow. Image Source: (Hashimoto et al., 2019).

To build such an evaluator to assess the quality of generated summaries, they introduce two components: (a) a set of relevant questions for each source document and (b) a question-answering system. They first generate questions from each reference summary by masking each of the named entities present in the reference based on the method described in Hermann et al. (2015). For each reference summary, this results in several triplets in the form *(generated summary, question, answer)*, where *question* refers to the sentence containing the masked entity, *answer* refers to the masked entity, and the *generated summary* is generated by their summarization model. Thus, for each generated summary, metrics can be derived based on the accuracy of the question answering system in retrieving the correct answers from each of the associated triplets. This metric is useful for summarizing documents for domains that contain lots of named entities, such as biomedical or news article summarization.

4.3 Regression-Based Evaluation

Shimanaka et al. (2018) propose a segment-level machine translation evaluation metric named **RUSE**. They treat the evaluation task as a regression problem to predict a scalar value to indicate the quality of translating a machine-translated hypothesis t to a reference translation r. They first do a forward pass on the GRU (gated-recurrent unit) based an encoder to generate t and represent r as a d-dimensional vector. Then, they apply different matching methods to extract relations between t and r by (1) concatenating $(\vec{t}, \vec{r})$; (2) getting the element-wise product $(\vec{t} * \vec{r})$; (3) computing the absolute element-wise distance $|\vec{t} - \vec{r}|$ (see Figure 4.5). RUSE is demonstrated to be an efficient metric in machine translation shared tasks in both segment-level (how well the metric correlates with human judgements of segment quality) and system-level (how well a given metric correlates with the machine translation workshop official manual ranking) metrics.

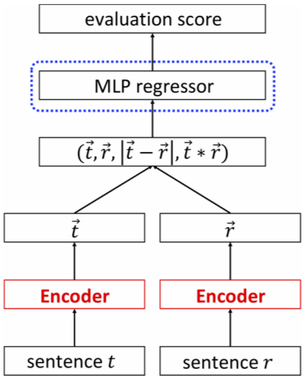


Figure 4.4: The sketch of the RUSE metric. Image source (Logeswaran & Lee, 2018).

4.4 Evaluation Models with Human Judgments

For more creative and open-ended text generation tasks, such as chit-chat dialog, story generation, or online review generation, current evaluation methods are only useful to some degree. As we mentioned in the beginning of this section, word-overlap metrics are ineffective as there are often many plausible references in these scenarios and collecting all is impossible. Even though human evaluation methods are useful in these scenarios for evaluating aspects like coherency, naturalness, or fluency, aspects like diversity or creativity may be difficult for human judges to assess as they have no knowledge about the dataset that the model is trained on. Language models can learn to copy from the training dataset and generate samples that a human judge will rate as high in quality, but will fail in generating diverse samples (i.e., samples that are very different from training samples), as has been observed in social chatbots (Li et al., 2016; Zhou et al., 2020). As we discussed in the previous sections, a language model optimized only for perplexity generates coherent but bland responses. Such behaviours are observed when generic pre-trained language models are used for downstream tasks 'as-is' without fine-tuning on in-domain datasets of related downstream tasks. A commonly overlooked issue is that conducting human evaluation for every new generation task can be expensive and not easily generalizable.

To calibrate human judgments and automatic evaluation metrics, model-based approaches that use human judgments as attributes or labels have been proposed. Lowe et al. (2017) introduce a model-based evaluation metric, ADEM, which is learned from human judgments for dialog system evaluation, specifically response generation in a chatbot setting. They collect human judgment scores on chitchat dialog using the "appropriateness" metric, which they say is satisfactory to evaluate chitchat dialog model responses as most systems generate inappropriate responses. They train the evaluation model using Twitter; each tweet response is a reference, and its previous dialog turns are its context. Then they use different models (such as RNNs, retrieval-based methods, or other human responses) to generate different responses and ask humans to judge the appropriateness of the generated response given the context. For evaluation they use a higher quality labeled Twitter dataset (Ritter et al., 2011), which contains dialogs on a variety of topics.

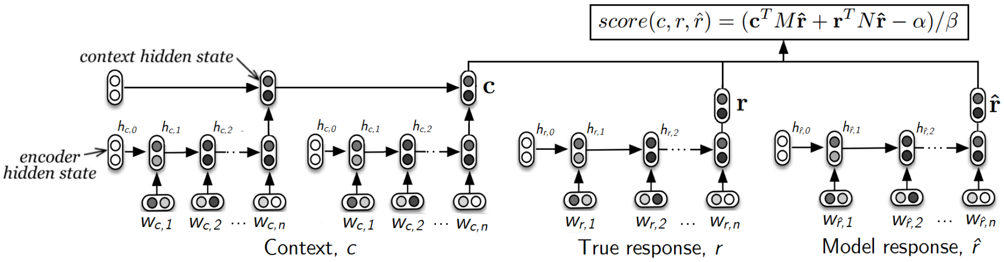

$score(c,r,\hat{r}) = (\mathbf{c}^T M\hat{\mathbf{r}} + \mathbf{r}^T N\hat{\mathbf{r}} - \alpha)/\beta$
context hidden state
encoder hidden state
c
r
$\hat{\mathbf{r}}$
$h_{c,0}$
$h_{c,1}$
$h_{c,2}$
$h_{r,0}$
$h_{r,1}$
$h_{r,2}$
$h_{\hat{r},0}$
$h_{\hat{r},1}$
$h_{\hat{r},2}$
$w_{c,1}$ $w_{c,2}$ ... $w_{c,n}$
$w_{r,1}$ $w_{r,2}$ ... $w_{r,n}$
$w_{\hat{r},1}$ $w_{\hat{r},2}$ ... $w_{\hat{r},n}$
Context, c
True response, r
Model response, $\hat{r}$

Figure 4.5: The ADEM evaluation model. Image source (Lowe et al., 2017).

Using this score-labeled dataset, the ADEM evaluation model is trained as follows: First, a latent variational recurrent encoder-decoder model (VHRED) (Serban et al., 2016b) is pre-trained on a dialog dataset to learn to represent the context of a dialog. VHRED encodes the dialog context into a vector representation, from which the model generates samples of initial vectors to condition the decoder model to generate the next response. Using the pre-trained VHRED model as the encoder, they train ADEM as follows. First, the dialog context, c, the model generated response $\hat{r}$, and the reference response r are fed to VHRED to get their embedding vectors, $\mathbf{c}$, $\hat{\mathbf{r}}$ and $\mathbf{r}$. Then, each embedding is linearly projected so that the model response $\hat{\mathbf{r}}$ can be mapped onto the spaces of the dialog context and the reference response to calculate a similarity score. The similarity score measures how close the model responses are to the context and the reference response after the projection, as follows:

$$score(c, \hat{r}, r) = (\mathbf{c}^T M \hat{\mathbf{r}} + \mathbf{r}^T N \hat{\mathbf{r}} - \alpha)/\beta \tag{4.1}$$

ADEM is optimized for squared error loss between the predicted score and the human judgment score with L-2 regularization in an end-to-end fashion. The trained evaluation model is shown to correlate well with human judgments. ADEM is also found to be conservative and give lower scores to plausible responses.

With the motivation that a good evaluation metric should capture both the quality and the diversity of the generated text, Hashimoto et al. (2019) propose a new evaluation metric named Human Unified with Statistical Evaluation (**HUSE**), which focuses on more creative and open-ended text generation tasks, such as dialogue and story generation. Different from the ADEM metric, which relies on human judgments for training the model, HUSE combines statistical evaluation and human evaluation metrics in one model, as shown in Figure 4.6.

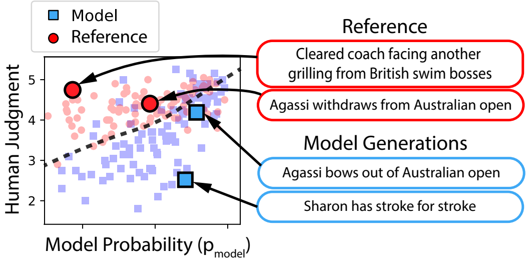


Figure 4.6: HUSE can identify samples with defects in quality (*Sharon has stroke for stroke*) and diversity (*Cleared coach facing*). Image Source: (Hashimoto et al., 2019).

HUSE considers the conditional generation task that, given a context x sampled from a prior distribution $p(x)$, outputs a distribution over possible sentences $p_{model}(y|x)$. The evaluation metric is designed to determine the similarity of the output distribution p_{model} and a human generation reference distribution p_{ref}. This similarity is scored using an *optimal discriminator* that determines whether a sample comes from the reference or hypothesis (model) distribution (Figure 4.6). For instance, a low-quality text is likely to be sampled from the model distribution. The discriminator is implemented approximately using two probability measures: (i) the probability of a sentence under the model, which can be estimated using the text generation model, and (ii) the probability under the reference distribution, which can be estimated based on human judgment scores. On summarization and chitchat dialog tasks, HUSE has been shown to be effective to detect low-diverse generations that humans fail to detect.

4.5 BERT-Based Evaluation

Given the strong performance of BERT (Devlin et al., 2018) across many tasks, there has been work that uses BERT or similar pre-trained language models for evaluating NLG tasks, such as summarization and dialog response generation. Here, we summarize some of the recent work that fine-tunes BERT to use as evaluation metrics for downstream text generation tasks.

One of the BERT-based models for semantic evaluation is **BERTSCORE** (Zhang et al., 2020). As illustrated in Figure 4.7, it leverages the pre-trained contextual embeddings from BERT and matches

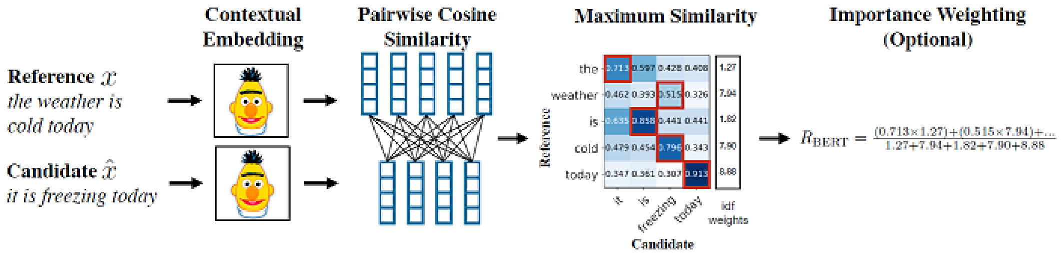


Figure 4.7: Illustration of BERTSCORE metric. Image Source: Zhang et al. (2020).

words in candidate and reference sentences by cosine similarity. It has been shown to correlate well with human judgments on sentence-level and system-level evaluations. Moreover, BERTSCORE computes precision, recall, and F1 measures, which are useful for evaluating a range of NLG tasks.

Kané et al. (2019) present a new BERT-based evaluation method called **ROBERTA-STS** to detect sentences that are logically contradictory or unrelated, regardless whether they are grammatically plausible. Using ROBERTA (Liu et al., 2019c) as a pre-trained language model, ROBERTA-STS is fine-tuned on the STS-B dataset to learn the similarity of sentence pairs on a Likert scale. Another evaluation model is fine-tuned on the Multi-Genre Natural Language Inference Corpus in a similar way to learn to predict logical inference of one sentence given the other. Both model-based evaluators have been shown to be more robust and correlate better with human evaluation than automatic evaluation metrics such as BLEU and ROUGE.

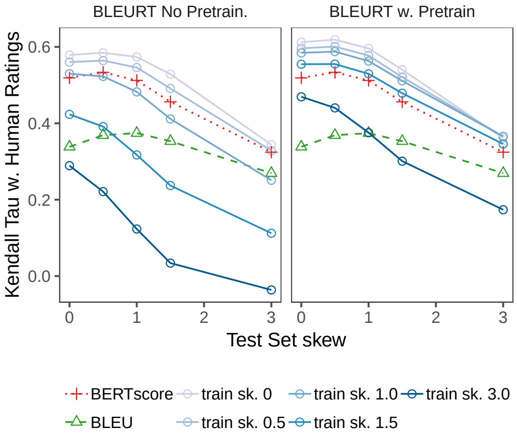


Figure 4.8: Agreement between BLEURT and human ratings for different skew factors in train and test. Image Source: Sellam et al. (2020)

Another recent BERT-based machine-learned evaluation metric is **BLEURT** (Sellam et al., 2020), which is proposed to evaluate various NLG systems. The evaluation model is trained as follows: A checkpoint from BERT is taken and fine-tuned on synthetically generated sentence pairs using automatic evaluation scores such as BLEU or ROUGE, and then further fine-tuned on system-generated outputs and human-written references using human ratings and automatic metrics as labels. The fine-tuning of BLEURT on synthetic pairs is an important step because it improves the robustness to quality drifts of generation systems.

As shown in the plots in Figure 4.8, as the NLG task gets more difficult, the ratings get closer as it is easier to discriminate between “good” and “bad” systems than to rank “good” systems. To ensure the robustness of their metric, they investigate with training datasets with different characteristics, such as when the training data is highly skewed or out-of-domain. They report that the training skew has a disastrous effect on BLEURT without pre-training; this pre-training makes BLEURT significantly more robust to quality drifts.

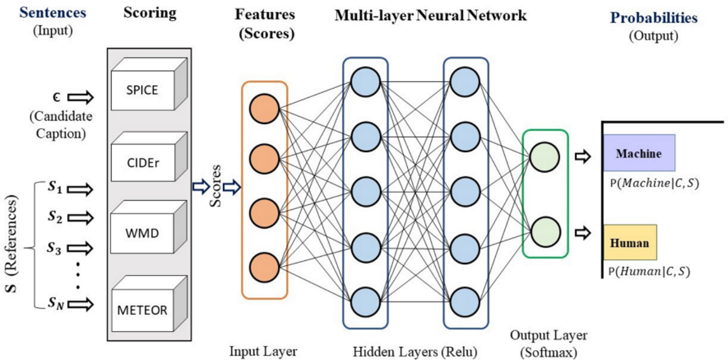


Figure 4.9: Composite Metrics model architecture. Image Source: (Sharif et al., 2018).

As discussed in Chapter 2, humans can efficiently evaluate performance of two models side-by-side, and most embedding-based similarity metrics reviewed in the previous sections are based on this idea. Inspired by this, the **comparator evaluator** (Zhou & Xu, 2020) is proposed to evaluate NLG models by learning to compare a pair of generated sentences by fine-tuning BERT. A text pair relation classifier is trained to compare the task-specific quality of a sample hypothesis and reference based on the win/loss rate. Using the trained model, a skill rating system is built. This system is similar to the player-vs-player games in which the players are evaluated by observing a record of wins and losses of multiple players. Then, for each player, the system infers the value of a latent, unobserved skill variable that indicates the records of wins and losses. On story generation and open domain dialogue response generation tasks, the comparator evaluator metric demonstrates high correlation with human evaluation.

4.6 Composite Metric Scores

The quality of many NLG models like machine translation and image captioning can be evaluated for multiple aspects, such as adequacy, fluency, and diversity. Many composite metrics have been proposed to capture a multi-dimensional sense of quality.

Sharif et al. (2018) present a machine-learned composite metric for evaluating image captions. The metric incorporates a set of existing metrics such as METEOR, WMD, and SPICE to measure both adequacy and fluency. Li & Chen (2020) propose a composite reward function to evaluate the performance of image captions. The approach is based on refined Adversarial Inverse Reinforcement Learning (rAIRL), which eases the reward ambiguity (common in reward-based generation models) by decoupling the reward for each word in a sentence. The proposed composite reward is shown on MS COCO data to achieve state-of-the-art performance on image captioning. Some examples generated from this model that uses the composite reward function are shown in

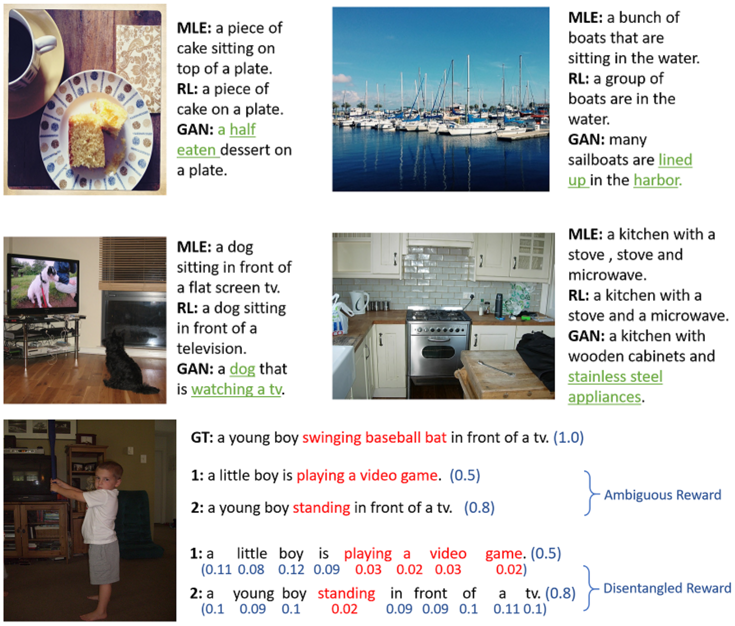

MLE: a piece of cake sitting on top of a plate.
RL: a piece of cake on a plate.
GAN: a half eaten dessert on a plate.
MLE: a bunch of boats that are sitting in the water.
RL: a group of boats are in the water.
GAN: many sailboats are lined up in the harbor.
MLE: a dog sitting in front of a flat screen tv.
RL: a dog sitting in front of a television.
GAN: a dog that is watching a tv.
MLE: a kitchen with a stove , stove and microwave.
RL: a kitchen with a stove and a microwave.
GAN: a kitchen with wooden cabinets and stainless steel appliances.
GT: a young boy swinging baseball bat in front of a tv. (1.0)
1: a little boy is playing a video game. (0.5)
2: a young boy standing in front of a tv. (0.8)
Ambiguous Reward
1: a little boy is playing a video game. (0.5)
(0.11 0.08 0.12 0.09 0.03 0.02 0.03 0.02)
2: a young boy standing in front of a tv. (0.8)
(0.1 0.09 0.1 0.02 0.09 0.09 0.1 0.11 0.1)
Disentangled Reward

Figure 4.10: (Top four images) Example image captions using different learning objectives: MLE: maximum likelihood learning, GAN: Generative Adversarial Networks, RL: Reward-based reinforcement learning. (Bottom image) Example generations from Adversarial Inverse Reinforcement Learning (rAIRL). Image Source: (Li & Chen, 2020).

Chapter 5

Two Case Studies of Task-Specific NLG Evaluation

In the previous chapters, we have reviewed a wide range of NLG evaluation metrics individually. This chapter presents how these metrics are used jointly to evaluate NLG systems for real-world applications. We choose two NLG tasks, automatic document summarization and long-text generation, as case studies. These tasks are sophisticated enough that multiple metrics are required to gauge different aspects of the NLG quality.

5.1 Case Study #1: Automatic Document Summarization Evaluation

A text summarization system aims to extract useful content from a reference document and generate a short summary that is coherent, fluent, readable, concise, and consistent with the reference document. There are different types of summarization approaches, which can be grouped by their tasks into (i) **generic** text summarization for broad topics; (ii) **topic-focused**, such as scientific article, conversation, or meeting summarization; and (iii) **query-focused**, such that the summary answers a posed query. These approaches can also be grouped by their method: (i) **extractive**, where a summary is composed of a subset of sentences or words in the input document; and (ii) **abstractive**, where a summary is generated on-the-fly and often contains text units that do not occur in the input document. Depending on the number of documents to be summarized, these approaches can be grouped into single-document or multi-document summarization.

Evaluation of text summarization, regardless of its type, measures the system's ability to generate a summary based on: (i) a set of criteria that are not related to references (Dusek et al., 2017), (ii) a set of criteria that measure its closeness to the reference document, or (iii) a set of criteria that measure its closeness to the reference summary. Figure 5.1 shows the taxonomy of evaluation metrics (Steinberger & Jezek, 2009) in two categories: intrinsic and extrinsic.

5.1.1 Intrinsic Methods

Intrinsic evaluation of generated summaries can focus on the generated text's content, text quality, and factual consistency, each discussed below.

Content. Content evaluation compares a generated summary to a reference summary using automatic metrics. The most widely used metric for summarization is ROUGE, though other metrics, such as BLEU and F-SCORE, are also used. Although ROUGE has been shown to correlate well with human judgments for generic text summarization, the correlation is lower for topic-focused summarization like extractive meeting summarization (Liu & Liu, 2008). Meetings are transcripts of spontaneous speech, and thus usually contain disfluencies, such as pauses (e.g., 'um,' 'uh,' etc.), discourse markers (e.g., 'you know,' 'i mean,' etc.), repetitions, etc. Liu & Liu (2008) find that after such disfluencies are cleaned, the ROUGE score is improved. They even observed fair amounts of

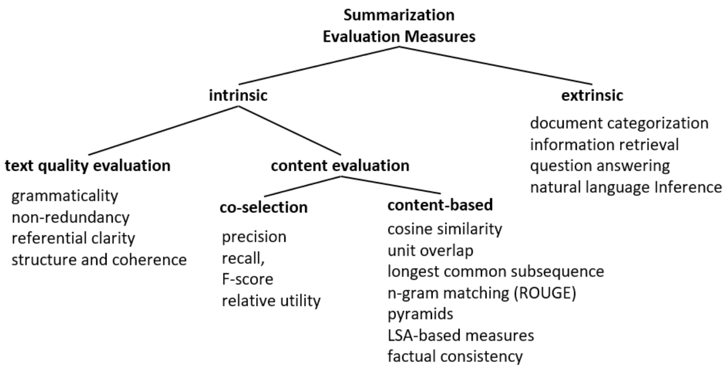


Figure 5.1: Taxonomy of summarization evaluation methods. Extended from Steinberger & Jezek (2009).

improvement in the correlation between the ROUGE score and human judgments when they include the speaker information of the extracted sentences from the source meeting to form the summary.

Quality. Evaluating generated summaries based on quality has been one of the challenging tasks for summarization researchers. As basic as it sounds, since the definition of a “good quality summary” has not been established, finding the most suitable metrics to evaluate quality remains an open research area. Below are some criteria of text, which are used in recent papers as human evaluation metrics to evaluate the quality of generated text in comparison to the reference text.

- **Coherence** measures how clearly the ideas are expressed in the summary (Lapata & Barzilay, 2005).
- **Readability and Fluency**, associated with non-redundancy, are linguistic quality metrics used to measure how repetitive the generated summary is and how many spelling and grammar errors there are in the generated summary. (Lapata, 2003).
- **Focus** indicates how many of the main ideas of the document are captured while avoiding superfluous details.
- **Informativeness**, which is mostly used to evaluate question-focused summarization, measures how well the summary answers a question. Auto-regressive generation models trained to generate a short summary text given a longer document(s) may yield shorter summaries due to reasons relating to bias in the training data or type of the decoding method (e.g., beam search can yield more coherent text compared to top-k decoding but can yield shorter text if a large beam size is used.) (Huang et al., 2017). Thus, in comparing different model generations, the summary text length has also been used as an informativeness measure since a shorter text typically preserves less information (Singh & Jin, 2016).

These quality criterion are widely used as evaluation metrics for human evaluation in document summarization. They can be used to compare a system-generated summary to a source text, a human-generated summary, or to another system-generated summary.

Factual Consistency. One thing that is usually overlooked in document summarization tasks is evaluating the generated summaries based on how well they can convey factual correctness. As discussed in the introduction section, fact checking has been a mainstream evaluation strategy for automatic text generation, due to the emergence of powerful language models (Zellers et al., 2019). However, since they are not trained to be factually consistent and can write about anything related to the prompt, they frequently generate factually incorrect text. Table 5.1 shows a sample summarization model output, in which the claims made are not consistent with the source document (Kryściński et al., 2019).

It is imperative that the summarization models are factually consistent and that any conflicts between a source document and its generated summary can be easily measured, especially for domain-specific summarization tasks like patient-doctor conversation summarization or business meeting

Source article fragments	
(CNN) The mother of a quadriplegic man who police say was left in the woods for days cannot be extradited to face charges in Philadelphia until she completes an unspecified "treatment," Maryland police said Monday. The Montgomery County (Maryland) Department of Police took Nyia Parler, 41, into custody Sunday (...)	(CNN) The classic video game "Space Invaders" was developed in Japan back in the late 1970's – and now their real-life counterparts are the topic of an earnest political discussion in Japan's corridors of power. Luckily, Japanese can sleep soundly in their beds tonight as the government's top military official earnestly revealed that (...)
Model generated claims	
Quadriplegic man Nyia Parler, 41, left in woods for days can not be extradited.	Video game "Space Invaders" was developed in Japan back in 1970.

Table 5.1: Examples of factually incorrect claims output by summarization models. Green text highlights the support in the source documents for the generated claims, red text highlights the errors made by summarization models. Table Source (Kryściński et al., 2019).

summarization. As a result, factual-consistency-aware text generation research has drawn a lot of attention in the community in recent years (Kryscinski et al., 2019; Gunel et al., 2019; Kryściński et al., 2019; Zhang et al., 2019b; Wang et al., 2020). A common approach is to use a model-based approach, in which a separate component is built on top of a summarization engine that can evaluate the generated summary based on factual consistency. In Section 4.2 we have discussed some of these parametric fact-checking models.

5.1.2 Extrinsic Summarization Evaluation Methods

Extrinsic evaluation metrics test the generated summary text by how it impacts the performance of downstream tasks, such as relevance assessment, reading comprehension, and question answering. Cohan & Goharian (2016) propose a new metric, SERA (Summarization Evaluation by Relevance Analysis), for summarization evaluation based on the content relevance of the generated summary and the human-written summary. They find that this metric yields higher correlation with human judgments compared to ROUGE, especially on the task of scientific article summarization. Eyal et al. (2019a); Wang et al. (2020) measure the performance of a summary by using it to answer a set of questions regarding the salient entities in the source document.

5.2 Case Study #2: Long Text Generation Evaluation

A long text generation system aims to generate multi-sentence text, such as a single paragraph or a multi-paragraph document. Common applications of long-form text generation are document-level machine translation, story generation, news article generation, poem generation, summarization, and image description generation, to name a few. This research area presents a particular challenge to state-of-the-art (SOTA) approaches that are based on statistical neural models, which are proven to be insufficient to generate coherent long text. For example, one of the SOTA neural language models, GPT-2 (Radford et al., 2018), can generate remarkably fluent sentences, and even paragraphs, for a given topic or a prompt. However, as more sentences are generated and the text gets longer, it starts to wander, switching to unrelated topics and becoming incoherent (Rashkin et al., 2020). Evaluating long-text generation by itself is a challenging task. New criteria need to be implemented to measure the quality of long generated text, such as inter-sentence or inter-paragraph coherence in language style and semantics. Although human evaluation methods are commonly used, we focus our discussion on automatic evaluation methods in this case study.

5.2.1 Evaluation via Discourse Structure

Long text consists of groups of sentences structured together by linguistic elements known as *discourse* (Jurafsky & Martin, 2009). Considering the discourse structure of the generated text is then crucial in evaluating the system. Especially in open-ended text generation, the model needs to determine the topical flow, structure of entities and events, and their relations in a narrative flow that is coherent and fluent. One of the major tasks in which discourse plays an important role is document-level machine translation (Gong et al., 2015). Hajlaoui & Popescu-Belis (2013) present a new metric called Accuracy of Connective Translation (ACT) (Meyer et al., 2012) that uses a combination of rules and automatic metrics to compare the discourse connection between the source and target documents. Joty et al. (2017), on the other hand, compare the source and target documents based on the similarity of their discourse trees.

5.2.2 Evaluation via Lexical Cohesion

Lexical cohesion is a surface property of text and refers to the way textual units are linked together grammatically or lexically. Lexical similarity (Lapata & Barzilay, 2005) is one of the most commonly used metrics in story generation. Roemmele et al. (2017) filter the n-grams based on lexical semantics and only use adjectives, adverbs, interjections, nouns, pronouns, proper nouns, and verbs for lexical similarity measure. Other commonly used metrics compare reference and source text on word- (Mikolov et al., 2013) or sentence-level (Kiros et al., 2015) embedding similarity averaged over the entire document. Entity co-reference is another metric that has been used to measure coherence (Elsner & Charniak, 2008). An entity should be referred to properly in the text and should not be used before introduced. Roemmele et al. (2017) capture the proportion of the entities in the generated sentence that are co-referred to an entity in the corresponding context as a metric of entity co-reference, in which a higher co-reference score indicates higher coherence.

In machine translation, Wong & Kit (2019) introduce a feature that can identify lexical cohesion at the sentence level via word-level clustering using WordNet (Miller, 1995) and stemming to obtain a score for each word token, which is averaged over the sentence. They find that this new score improves correlation of BLEU and TER with human judgments. Other work, such as Gong et al. (2015), uses topic modeling together with automatic metrics like BLEU and METEOR to evaluate lexical cohesion in machine translation of long text. Chow et al. (2019) investigate the position of the word tokens in evaluating the *fluency* of the generated text. They modify WMD by adding a fragmentation penalty to measure the fluency of a translation for evaluating machine translation systems.

5.2.3 Evaluation via Writing Style

Gamon (2004) show that an author's writing is consistent in style across a particular work. Based on this finding, Roemmele et al. (2017) propose to measure the quality of generated text based on whether it presents a consistent writing style. They capture the category distribution of individual words between the story context and the generated following sentence using their part-of-speech tags of words (e.g, adverbs, adjectives, conjunctions, determiners, nouns, etc.).

Text style transfer reflects the creativity of the generation model in generating new content. Style transfer can help re-write a text in a different style, which is useful in creative writing such as poetry generation (Ghazvininejad et al., 2016). One metric that is commonly used in style transfer is the classification score obtained from a pre-trained style transfer model (Fu et al., 2018). This metric measures whether a generated sentence has the same style as its context.

5.2.4 Evaluation with Multiple References

One issue of evaluating text generation systems is the diversity of generation, especially when the text to evaluate is long. The generated text can be fluent, valid given the input, and informative for the user, but it still may not have lexical overlap with the reference text or the prompt that was used to constrain the generation. This issue has been investigated extensively (Li et al., 2016; Montahaei et al., 2019b; Holtzman et al., 2020; Welleck et al., 2019; Gao et al., 2019). Using multiple references that cover as many plausible outputs as possible is an effective solution to improving the correlation of automatic evaluation metrics (such as adequacy and fluency) with human judgments, as demonstrated in machine translation (Han, 2018; Läubli et al., 2020) and other NLG tasks.

Chapter 6

Conclusions and Future Directions

Text generation is central to many NLP tasks, including machine translation, dialog response generation, document summarization, etc. With the recent advances in neural language models, the research community has made significant progress in developing new NLG models and systems for challenging tasks like multi-paragraph document generation or visual story generation. With every new system or model comes a new challenge of evaluation. This paper surveys the NLG evaluation methods in three categories:

- **Human-Centric Evaluation.** Human evaluation is the most important for developing NLG systems and is considered the gold standard when developing automatic metrics. But it is expensive to execute, and the evaluation results are difficult to reproduce.
- **Untrained Automatic Metrics.** Untrained automatic evaluation metrics are widely used to monitor the progress of system development. A good automatic metric needs to correlate well with human judgments. For many NLG tasks, it is desirable to use multiple metrics to gauge different aspects of the system's quality.
- **Machine-Learned Evaluation Metrics.** In the cases where the reference outputs are not complete, we can train an evaluation model to mimic human judges. However, as pointed out in Gao et al. (2019), any machine-learned metrics might lead to potential problems such as overfitting and 'gaming of the metric.'

We conclude this paper by summarizing some of the challenges of evaluating NLG systems:

- **Detecting machine-generated text and fake news.** As language models get stronger by learning from increasingly larger corpora of human-written text, they can generate text that is not easily distinguishable from human-authored text. Due to this, new systems and evaluation methods have been developed to detect if a piece of text is machine- or human-generated. A recent study (Schuster et al., 2019) reports the results of a fact verification system to identify inherent bias in training datasets that cause fact-checking issues. In an attempt to combat fake news, Vo & Lee (2019) present an extensive analysis of tweets and a new tweet generation method to identify fact-checking tweets (among many tweets), which were originally produced to persuade posters to stop tweeting fake news. Other research focuses on factually correct text generation, with a goal of providing users with accurate information. Massarelli et al. (2019) introduce a new approach for generating text that is factually consistent with the knowledge source. Kryściński et al. (2019) investigate methods of checking the consistency of a generated summary against the document from which the summary is generated.
- **Making evaluation explainable.** Explainable AI refers to AI and machine learning methods that can provide human-understandable justifications for their behaviour (Ehsan et al., 2019). Evaluation systems that can provide reasons for their decisions are beneficial in many ways. For instance, the explanation could help system developers to identify the root causes of the system's quality problems such as unintentional bias, repetition, or factual inconsistency. The field of explainable AI is growing, particularly in generating explanations

of classifier predictions in NLP tasks (Ribeiro et al., 2016, 2018; Thorne et al., 2019). Text generation systems that use evaluation methods that can provide justification or explanation for their decisions will be more trusted by their users. Future NLG evaluation research should focus on developing easy-to-use, robust, and explainable evaluation tools.

- **Improving corpus quality.** Creating high-quality datasets with multiple reference texts is essential for not only improving the reliability of evaluation but also allowing the development of new automatic metrics that correlate well with human judgments (Belz & Reiter, 2006).
- **Standardizing evaluation methods.** Most untrained automatic evaluation metrics are standardized using open source platforms like Natural Language Toolkit (NLTK)[1] or spaCy[2]. Such platforms can significantly simplify the process of benchmarking different models. However, there are still many NLG tasks that use task-specific evaluation metrics, such as metrics to evaluate the contextual quality or informativeness of generated text. There are also no standard criteria for human evaluation methods for different NLG tasks.

  It is important for the research community to collaborate more closely to standardize the evaluation metrics for NLG tasks that are pursued by many research teams. One effective way to achieve this is to organize challenges or shared tasks, such as the Evaluating Natural Language Generation Challenge[3] and the Shared Task on NLG Evaluation[4].
- **Developing effective human evaluations.** For most NLG tasks, there is little consensus on how human evaluations should be conducted. Furthermore, papers often leave out important details on how the human evaluations were run, such as who the evaluators are and how many people evaluated the text (van der Lee et al., 2019). Clear reporting of human evaluations is very important, especially for replicability purposes.

  We encourage NLG researchers to design their human evaluations carefully, paying attention to best practices described in NLG and crowdsourcing research, and to include the details of the studies and data collected from human evaluations, where possible, in their papers. This will allow new research to be consistent with previous work and enable more direct comparisons between NLG results. Human evaluation-based shared tasks and evaluation platforms can also provide evaluation consistency and help researchers directly compare how people perceive and interact with different NLG systems.
- **Evaluating ethical issues.** There is still a lack of systematic methods for evaluating how effectively an NLG system can avoid generating improper or offensive language. The problem is particularly challenging when the NLG system is based on neural language models whose output is not always predictable. As a result, many social chatbots, such as XiaoIce (Zhou et al., 2020), resort to hand-crafted policies and editorial responses to make the system's behavior predictable. However, as pointed out by Zhou et al. (2020), even a completely deterministic function can lead to unpredictable behavior. For example, a simple answer "Yes" could be perceived as offensive in a given context.

We encourage researchers working in NLG and NLG evaluation to focus on these challenges moving forward, as they will help sustain and broaden the progress we have seen in NLG so far.

[1] nltk.org

[2] spacy.io

[3] https://framalistes.org/sympa/info/eval.gen.chal

[4] https://github.com/evanmiltenburg/Shared-task-on-NLG-Evaluation

Bibliography

Abhaya Agarwal and Alon Lavie. Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with human rankings of machine translation output. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pp. 115–118, USA, 2008. Association for Computational Linguistics. ISBN 9781932432091.

Eneko Agirre, Aitor Gonzalez-Agirre, Inigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. Semeval-2016 task 2: Interpretable semantic textual similarity. pp. 512–524, 01 2016.

Manex Agirrezabal, Bertol Arrieta, Aitzol Astigarraga, and Mans Hulden. POS-tag based poetry generation with WordNet. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pp. 162–166, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W13-2121.

Jorge Agnese, Jonathan Herrera, Haicheng Tao, and Xingquan Zhu. A survey and taxonomy of adversarial neural networks for text-to-image synthesis. *WIREs Data Mining and Knowledge Discovery*, Feb 2020. ISSN 1942-4795. doi: 10.1002/widm.1345. URL http://dx.doi.org/10.1002/widm.1345.

Ahmed Ali, Walid Magdy, Peter Bell, and Steve Renais. Multi-reference wer for evaluating asr for languages with no orthographic rules. pp. 576–580, 12 2015. doi: 10.1109/ASRU.2015.7404847.

Ramiz Aliguliyev. Using the f-measure as similarity measure for automatic text summarization. *Vychislitel'nye Tekhnologii*, 13, 01 2008.

Jacopo Amidei, Paul Piwek, and Alistair Willis. Rethinking the agreement in human evaluation tasks. In *COLING*, 2018.

Jacopo Amidei, Paul Piwek, and Alistair Willis. Agreement is overrated: A plea for correlation to assess human evaluation reliability. In *INLG*, 2019a.

Jacopo Amidei, Paul Piwek, and Alistair Willis. The use of rating and Likert scales in Natural Language Generation human evaluation tasks: A review and some recommendations. *INLG*, 2019b. URL https://www.inlg2019.com/assets/papers/57_Paper.pdf.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: semantic propositional image caption evaluation. *ECCV*, 2016. URL http://arxiv.org/abs/1607.08822.

Erion Çano and Ondřej Bojar. Keyphrase generation: A multi-aspect survey. 2019.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. January 2019. 5th International Conference on Learning Representations, ICLR 2017 ; Conference date: 24-04-2017 Through 26-04-2017.

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34:555–596, 2008.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Shuang Bai and Shan An. Va survey on automatic image caption generation. *Neuro Computing*, pp. 291–304, 10 2018.

Mousumi Banerjee, Michelle Hopkins Capozzoli, Laura A. McSweeney, and Debajyoti Sinha. Beyond kappa: A review of interrater agreement measures. 1999.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W05-0909`.

Ellen Gurman Bard, Dan Robertson, and Antonella Sorace. Magnitude estimation of linguistic acceptability. *Language*, 72(1):32–68, 1996. ISSN 00978507, 15350665. URL `http://www.jstor.org/stable/416793`.

Regina Barzilay and Lillian Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003: Main Proceedings*, pp. 16–23, 2003.

Anja Belz and Ehud Reiter. Comparing automatic and human evaluation of nlg systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/E06-1040`.

Alan Black, Susanne Burger, Alistair Conkie, Helen Hastie, Simon Keizer, Oliver Lemon, Nicolas Merigaud, Gabriel Parent, Gabriel Schubiner, Blaise Thomson, Jason Williams, Kai Yu, Steve Young, and Maxine Eskenazi. Spoken dialog challenge 2010: Comparison of live and control test results. pp. 2–7, 01 2011.

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. Discourse-aware neural rewards for coherent text generation. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT 2018)*, July 2018.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language gans falling short. *CoRR*, abs/1811.02549, 2018. URL `http://arxiv.org/abs/1811.02549`.

Chris Callison-burch and Miles Osborne. Re-evaluating the role of bleu in machine translation research. In *In EACL*, pp. 249–256, 2006.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Meta-evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 136–158, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W07-0718`.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. July 2018.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055, 2017. URL `http://arxiv.org/abs/1708.00055`.

Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. Adversarial text generation via feature mover's distance. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 4666–4677. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/7717-adversarial-text-generation-via-feature-movers-distance.pdf`.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1657–1668, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1152. URL `https://www.aclweb.org/anthology/P17-1152`.

Yen-Chun Chen and Mohit Bansal. 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)*, pp. 675–686, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1063. URL `https://www.aclweb.org/anthology/P18-1063`.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D14-1179.

Julian Chow, Lucia Specia, and Pranava Madhyastha. WMDO: Fluency-based word mover's distance for machine translation evaluation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 494–500, Florence, Italy, August 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-5356.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *ACL*, 2019.

J. Clarke and M. Lapata. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429, 2008.

Arman Cohan and Nazli Goharian. Revisiting summarization evaluation for scientific articles. *CoRR*, abs/1604.00400, 2016. URL http://arxiv.org/abs/1604.00400.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–, 04 1960.

Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220, 1968. doi: 10.1037/h0026256. URL https://doi.org/10.1037/h0026256.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1070.

Scott A. Crossley, Minkyung Kim, Laura Allen, and Danielle McNamara. Automated summarization evaluation (ase) using natural language processing tools. In *Artificial Intelligence in Education - 20th International Conference, AIED 2019, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 84–95. Springer Verlag, 2019.

Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge J. Belongie. Learning to evaluate image captioning. *CoRR*, abs/1806.06422, 2018. URL http://arxiv.org/abs/1806.06422.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. A comprehensive survey of multilingual neural machine translation, 2020.

Daniel Dahlmeier, Chang Liu, and Hwee Tou Ng. Tesla at wmt 2011: Translation evaluation and tunable metric, 2011.

Andrew M. Dai, Christopher Olah, and Quoc V. Le. Document embedding with paragraph vectors. In *NeurIPS Deep Learning Workshop*, 2015.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *ICLR*, 2020.

Luciano Del Corro and Rainer Gemulla. Clausie: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pp. 355–366, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320351. doi: 10.1145/2488388.2488420. URL https://doi.org/10.1145/2488388.2488420.

Etienne Denoual and Yves Lepage. BLEU in characters: Towards automatic MT evaluation in languages without word delimiters. In *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*, 2005. URL https://www.aclweb.org/anthology/I05-2014.

Jan Deriu, Álvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. Evaluation metrics for text summarization. *Computing and Informatics*, 28/2, 2009. URL http://www.cai.sk/ojs/index.php/cai/article/viewFile/37/24.

Jan Deriu, Álvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. Survey on evaluation methods for dialogue systems. *CoRR*, abs/1905.04071, 2019. URL http://arxiv.org/abs/1905.04071.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Djellel Eddine Difallah, Elena Filatova, and Panagiotis G. Ipeirotis. Demographics and dynamics of mechanical turk workers. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018.

George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. pp. 138–145, 01 2002.

Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing, January 2005. URL https://www.microsoft.com/en-us/research/publication/automatically-constructing-a-corpus-of-sentential-paraphrases/.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *CoRR*, abs/1905.03197, 2019. URL http://arxiv.org/abs/1905.03197.

Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. *CoRR*, abs/1705.00106, 2017. URL http://arxiv.org/abs/1705.00106.

Ondrej Dusek, Jekaterina Novikova, and Verena Rieser. Referenceless quality estimation for natural language generation. *CoRR*, abs/1708.01759, 2017. URL http://arxiv.org/abs/1708.01759.

Ondrej Dusek, Jekaterina Novikova, and Verena Rieser. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG challenge. *CoRR*, abs/1901.07931, 2019. URL http://arxiv.org/abs/1901.07931.

Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O. Riedl. Automated rationale generation: A technique for explainable ai and its effects on human perceptions. *Proceedings of the 24th International Conference on Intelligent User Interfaces*, Mar 2019. doi: 10.1145/3301275.3302316. URL http://dx.doi.org/10.1145/3301275.3302316.

M. Elsner and E. Charniak. Coreference-inspired coherence modeling. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, 2008.

Matan Eyal, Tal Baumel, and Michael Elhadad. Question answering as an automatic evaluation metric for news article summarization. *Proceedings of the 2019 Conference of the North*, 2019a. URL http://dx.doi.org/10.18653/v1/n19-1395.

Matan Eyal, Tal Baumel, and Michael Elhadad. Question answering as an automatic evaluation 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)*, pp. 3938–3948, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1395.

Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical neural story generation. *CoRR*, abs/1805.04833, 2018. URL http://arxiv.org/abs/1805.04833.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: long form question answering. *CoRR*, abs/1907.09190, 2019a. URL http://arxiv.org/abs/1907.09190.

Angela Fan, Mike Lewis, and Yann N. Dauphin. Strategies for structuring story generation. *CoRR*, abs/1902.01109, 2019b. URL http://arxiv.org/abs/1902.01109.

JL Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5): 378–382, November 1971. ISSN 0033-2909. doi: 10.1037/h0031619. URL https://doi.org/10.1037/h0031619.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

Michael Gamon. Linguistic correlates of style: authorship classification with deep linguistic analysis features. *Association for Computational Linguistics*, 2004.

Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298, 2019.

Cristina Garbacea, Samuel Carton, Shiyan Yan, and Qiaozhu Mei. Judge the judges: A large-scale evaluation study of neural language models for online review generation. *CoRR*, abs/1901.00398, 2019. URL `http://arxiv.org/abs/1901.00398`.

Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.*, 61:65–170, 2017.

M. Ghazvininejad, X. Shi, Y. Choi, and K. Knight. Generating topical poetry. *EMNLP*, 2016.

Dimitra Gkatzia and Saad Mahamood. A snapshot of NLG evaluation practices 2005 - 2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pp. 57–60, Brighton, UK, September 2015. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W15-4708`.

Yoav Goldberg, Graeme Hirst, Yang Liu, , and Meng Zhang. Neural network methods for natural language processing. In *Computational Linguistics*, volume 44(1), 2018.

Zhengxian Gong, Min Zhang, and Guodong Zhou. Document-level machine translation evaluation with gist consistency and text cohesion. pp. 33–40, 2015.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv 1406.2661*, 2014.

Cyril Goutte. Automatic evaluation of machine translation quality. 2006.

Yvette Graham. Re-evaluating automatic summarization with BLEU and 192 shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 128–137, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D15-1013`.

Yvette Graham and Timothy Baldwin. Testing for significance of increased correlation with human judgment. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 172–176, Doha, Qatar, October 2014. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D14-1020`.

Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013. URL `http://arxiv.org/abs/1308.0850`.

Beliz Gunel, Chenguang Zhu, Michael Zeng, and Xuedong Huang. Mind the facts: Knowledge-boosted coherent abstractive text summarization. *Knowledge Representation and Reasoning Meets Machine Learning Workshop at NeurIPS*, 2019.

Najeh Hajlaoui and Andrei Popescu-Belis. Assessing the accuracy of discourse connective translations: Validation of an automatic metric. *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing*, 2013.

Aaron L.-F Han and Derek Wong. Machine translation evaluation: A survey. *https://arxiv.org/abs/1605.04515*, 05 2016.

Aaron L.-F Han, Derek Wong, Lidia Chao, Liangye He, and Yi Lu. Unsupervised quality estimation model for english to german translation and its application in extensive supervised evaluation. *The Scientific World Journal*, 2013a.

Aaron L.-F Han, Derek Wong, Lidia Chao, Liangye He, Yi Lu, Junwen Xing, and Xiaodong Zeng. Mt summit13.language-independent model for machine translation evaluation with reinforced factors. 09 2013b.

Lifeng Han. Machine translation evaluation resources and methods: A survey. *IPRC-2018 - Ireland Postgraduate Research Conference*, 2018.

Tatsunori Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation. 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)*, pp. 1689–1701, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N19-1169`.

Helen Hastie and Anja Belz. A comparative evaluation methodology for NLG in interactive systems. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*

(LREC'14), pp. 4004–4011, Reykjavik, Iceland, May 2014a. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2014/pdf/1147_Paper.pdf`.

Helen F. Hastie and Anja Belz. A comparative evaluation methodology for nlg in interactive systems. In *LREC*, 2014b.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340, 2015. URL `http://arxiv.org/abs/1506.03340`.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017. URL `http://arxiv.org/abs/1706.08500`.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. *CoRR*, abs/1602.03483, 2016. URL `http://arxiv.org/abs/1602.03483`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *ICLR*, abs/1904.09751, 2020.

Md. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *CoRR*, abs/1810.04020, 2018. URL `http://arxiv.org/abs/1810.04020`.

Liang Huang, Kai Zhao, and Mingbo Ma. When to finish? optimal beam search for neural text generation (modulo beam size). *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017. doi: 10.18653/v1/d17-1227. URL `http://dx.doi.org/10.18653/v1/D17-1227`.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using click-through data. ACM International Conference on Information and Knowledge Management (CIKM), October 2013. URL `https://www.microsoft.com/en-us/research/publication/learning-deep-structured-semantic-models-for-web-search-using-clickthrough-data/`.

Xuedong Huang, Fileno Alleva, Hsiao wuen Hon, Mei yuh Hwang, and Ronald Rosenfeld. The sphinx-ii speech recognition system: An overview. *Computer, Speech and Language*, 7:137–148, 1992.

Text Inspector. Measure lexical diversity, 2013. URL `https://textinspector.com/help/lexical-diversity/`.

Panagiotis G. Ipeirotis, F. Provost, and Jing Wang. Quality management on amazon mechanical turk. In *HCOMP '10*, 2010.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 944–952, Cambridge, MA, October 2010. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D10-1092`.

Ming Jiang, Qiuyuan Huang, Lei Zhang, Xin Wang, Pengchuan Zhang, Zhe Gan, Jana Diesner, and Jianfeng Gao. Tiger: Text-to-image grounding for image caption evaluation. In *EMNLP 2019*, November 2019.

Shafiq Joty, Francisco Guzman, Lluis Marquez, and Preslav Nakov. Discourse structure in machine translation evaluation. *Computational Linguistics*, 43(4):683–722, 2017.

Daniel Jurafsky and James H. Martin. Asking and answering questions to evaluate the factual consistency of summaries. 2009.

Filip Jurcícek, Simon Keizer, Milica Gasic, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. Real user evaluation of spoken dialogue systems using amazon mechanical turk. pp. 3061–3064, 01 2011.

Sushant Kafle and Matt Huenerfauth. Evaluating the usability of automatically generated captions for people who are deaf or hard of hearing. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 165–174, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349260.

Hassan Kané, Yusuf Kocyigit, Pelkins Ajanoh, Ali Abdalla, and Mohamed Coulibali. Towards neural language evaluators. *Neurips 2019 Document Intelligence Workshop*, 2019.

David Kauchak and Regina Barzilay. Paraphrasing for automatic evaluation. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 2006.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. Re-evaluating automatic metrics for image captioning. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017. doi: 10.18653/v1/e17-1019. URL `http://dx.doi.org/10.18653/v1/e17-1019`.

Yoon Kim, Sam Wiseman, and Alexander M. Rush. A tutorial on deep latent variable models of natural language. *CoRR*, abs/1812.06834, 2018. URL `http://arxiv.org/abs/1812.06834`.

Svetlana Kiritchenko and Saif M. Mohammad. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 811–817, San Diego, California, June 2016. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N16-1095`.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. *CoRR*, abs/1506.06726, 2015. URL `http://arxiv.org/abs/1506.06726`.

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 423–430, Sapporo, Japan, July 2003. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P03-1054`.

D. Knight, K.—Marcu. Statistics-based summarization – step one: Sentence compression. *In Proceeding of The 17th National Conference of the American Association for Artificial Intelligence*, pp. 703–710, 2000.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text generation from knowledge graphs with graph transformers. *ArXiv*, abs/1904.02342, 2019.

E. Krahmer and M. Theune. *Empirical Methods in Natural Language Generation: Data-oriented Methods and Empirical Evaluation*. LNCS sublibrary: Artificial intelligence. Springer, 2010. ISBN 9783642155727. URL `https://books.google.com/books?id=aifpm9shAw8C`.

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

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. 2019.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 540–551, Hong Kong, China, November 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D19-1051`.

M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In *ICML*, 2015.

Lori Lamel, Sophie Rosset, Jean-Luc Gauvain, Samir Bennacef, Matine Garnier-Rizet, and Bernard Prouts. The limsi arise system. In *Speech Communication*, pp. 339–353, 2000.

Weiyu Lan, Xirong Li, and Jianfeng Dong. Fluency-guided cross-lingual image captioning. *ACL Multimedia*, abs/1708.04390, 2017. URL `http://arxiv.org/abs/1708.04390`.

Mirealla Lapata. Probabilistic text structuring: Experiments with sentence ordering. *proceedings of the annual meeting of the Association for Computational Linguistics, The Association of Computational Linguistics*, 2003.

Mirealla Lapata and Regina Barzilay. Automatic evaluation of text coherence: Models and representations. *In Kaelbling, L.P., Saffiotti, A., eds.: IJCAI, Professional Book Center*, 2005.

Alex Lascarides and Nicholas Asher. Discourse relations and defeasible knowledge. In *29th Annual Meeting of the Association for Computational Linguistics*, pp. 55–62, Berkeley, California, USA, June 1991. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P91-1008.

Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pp. 228–231, USA, 2007. Association for Computational Linguistics.

Alon Lavie, Kenji Sagae, and Shyamsundar Jayaraman. The significance of recall in automatic metrics for mt evaluation. In *AMTA*, 2004.

Audrey J. Lee and Mark A. Przybocki. Nist 2005 machine translation evaluation official results. 2005.

Chris Van Der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. Best practices for the human evaluation of automatically generated text. *INLG*, 2019. URL https://www.inlg2019.com/assets/papers/98_Paper.pdf.

Gregor Leusch, Nicola Ueffing, and Hermann Ney. CDER: Efficient MT evaluation using block movements. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E06-1031.

Jinchao Li, Qi Zhu, Baolin Peng, Lars Liden, Runze Liang, Ryuichi Takanobu, Shahin Shayandeh, Swadheen Shukla, Zheng Zhang, Minlie Huang, and Jianfeng Gao. Multi-domain task-oriented dialog challenge ii. 2020.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, June 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N16-1014.

Nannan Li and Zhenzhong Chen. Learning compact reward for image captioning. *arXiv 2003.10925*, 2020.

Sheng Li, Zhiqiang Tao, and Yun Fu. Visual to text: Survey of image and video captioning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, PP:1–16, 01 2019. doi: 10.1109/TETCI.2019.2892755.

Zhongyang Li, Xiao Ding, and Ting Liu. Generating reasonable and diversified story ending using sequence to sequence model with adversarial training. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1033–1043, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1088.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 605–612, Barcelona, Spain, July 2004. URL https://www.aclweb.org/anthology/P04-1077.

Dahua Lin, Sanja Fidler, and Chen Kong Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *In CVPR*, 2014.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2122–2132, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D16-1230.

Ding Liu and Daniel Gildea. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 25–32, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W05-0904`.

Feifan Liu and Yang Liu. Correlation between rouge and human evaluation of extractive meeting summaries. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pp. 201–204, USA, 2008. Association for Computational Linguistics.

Lixin Liu, Jiajun Tang, Xiaojun Wan, and Zongming Guo. Generating diverse and descriptive image captions using visual paraphrases. 2019a.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. pp. 873–881, 10 2017.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *ACL 2019*, June 2019b. URL `https://www.microsoft.com/en-us/research/publication/multi-task-deep-neural-networks-for-natural-language-understanding-2/`.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019c. URL `http://arxiv.org/abs/1907.11692`.

Chi-Kiu Lo. Meant 2.0: Accurate semantic mt evaluation for any output language. In *WMT*, 2017.

Chi-kiu Lo. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 507–513, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5358. URL `https://www.aclweb.org/anthology/W19-5358`.

Chi-Kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. Fully automatic semantic mt evaluation. In *WMT@NAACL-HLT*, 2012.

Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJvJXZb0W`.

Dang Hoang Long, Minh-Tien Nguyen, Ngo Xuan Bach, Le-Minh Nguyen, and Tu Minh Phuong. An entailment-based scoring method for content selection in document summarization. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, pp. 122–129, New York, NY, USA, 2018. Association for Computing Machinery.

Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press, 2015.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. Towards an automatic turing test: Learning to evaluate dialogue responses. *ACL*, 2017. URL `http://arxiv.org/abs/1708.07149`.

Sidi Lu, Yaoming Zhu, Weinan Zhang, Jun Wang, and Yong Yu. Neural text generation: Past, present and beyond. *CoRR*, abs/1803.07133, 2018. URL `http://arxiv.org/abs/1803.07133`.

Samuel Läubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. A set of recommendations for assessing human–machine parity in language translation. *Journal of Artificial Intelligence Research*, 67, 03 2020.

Kelvin Luu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith. Citation text generation. *ArXiv*, abs/2002.00317, 2020.

Nabin Maharjan, Rajendra Banjade, Dipesh Gautam, Lasang J. Tamang, and Vasile Rus. DT_Team at SemEval-2017 task 1: Semantic similarity using alignments, sentence-level embeddings and Gaussian mixture model output. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada, August 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/S17-2014`.

W.C. Mann and S.A. Thompson. Rhetorical structure theory: Description and construction of text structures. *In: Kempen G. (eds) Natural Language Generation. NATO ASI Series (Series E: Applied Sciences)*, 135, 1987.

Daniel Marcu. From discourse structures to text summaries. *Proceedings of ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pp. 80–88, 1997.

A. Martin and M. Przybocki. The nist 1999 speaker recognition evaluation - an overview, 2000.

Lara J. Martin, Prithviraj Ammanabrolu, William Hancock, Shruti Singh, Brent Harrison, and Mark O. Riedl. Event representations for automated story generation with deep neural nets. *CoRR*, abs/1706.01331, 2017. URL `http://arxiv.org/abs/1706.01331`.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. How decoding strategies affect the verifiability of generated text, 2019.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2799–2808, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1269. URL `https://www.aclweb.org/anthology/P19-1269`.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics. *Association for Computational Linguistics (ACL 2020)*, 2020.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. On faithfulness and factuality in abstractive summarization. *ArXiv*, abs/2005.00661, 2020.

P.M. McCarthy and S. Jarvis. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaces to lexical diversity assessment. In *Behaviour Research Methods*, volume 42(2), pp. 381–392, 2010. URL `https://link.springer.com/article/10.3758/BRM.42.2.381`.

Iain Mccowan, Darren Moore, John Dines, Daniel Gatica-Perez, Mike Flynn, Pierre Wellner, and Herve Bourlard. On the use of information retrieval measures for speech recognition evaluation. 01 2004.

Kathleen R. McKeown. Text generation. using discourse strategies and focus constraints to generate natural language text. *Studies in natural language processing*, 1985.

I. Melamed, Ryan Green, and Joseph Turian. Precision and recall of machine translation. 2003.

Thomas Meyer, Andrei Popescu-Belis, Najeh Hajlaoui, and Andrea Gesmundo. Machine translation of labeled discourse connectives. *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas*, 2012.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013. URL `http://arxiv.org/abs/1310.4546`.

George A. Miller. Wordnet: A lexical database for english. *Association for Computing Machinery*, 38(11):39–41, November 1995.

Tanushree Mitra, Clayton J. Hutto, and Eric Gilbert. Comparing person- and process-centric strategies for obtaining quality data on amazon mechanical turk. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. Jointly measuring diversity and quality in text generation models. *CoRR*, abs/1904.03971, 2019a. URL `http://arxiv.org/abs/1904.03971`.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. Jointly measuring diversity and quality in text generation models. *NeuralGen Workshop at NAACL 2019*, abs/1904.03971, 2019b. URL `http://arxiv.org/abs/1904.03971`.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. July 2018.

Preksha Nema and Mitesh M. Khapra. Towards a better metric for evaluating question generation systems. *CoRR*, abs/1808.10192, 2018. URL `http://arxiv.org/abs/1808.10192`.

Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. pp. 145–152, 01 2004.

Austin Lee Nichols and Jon K. Maner. The good-subject effect: investigating participant demand characteristics. *The Journal of general psychology*, 135 2:151–65, 2008.

Sonja Nießen, Franz Josef Och, Gregor Leusch, and Hermann Ney. An evaluation tool for machine translation: Fast evaluation for MT research. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece, May 2000. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2000/pdf/278.pdf`.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2241–2252, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D17-1238`.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. RankME: Reliable human ratings for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 72–78, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N18-2012`.

Kenji Ono, Kazuo Sumita, and Seiji Miike. Abstract generation based on rhetorical structure extraction. *Proceedings of the International Conference on Computational Linguistics (COLING'94)*, 1994.

Daniel M Oppenheimer, Tom Meyvis, and Nicolas Davidenko. Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of experimental social psychology*, 45 (4):867–872, 2009.

Martin T. Orne. On the social psychology of the psychological experiment: With particular reference to demand characteristics and their implications. 1962.

Sebastian Padó, Michel Galley, Dan Jurafsky, and Christoper Manning. Textual entailment features for machine translation evaluation. pp. 37–41, 01 2009.

Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. Recent advances in neural question generation. 2019. URL `http://arxiv.org/abs/1905.08949`.

Joybrata Panja and Sudip Kumar Naskar. ITER: Improving translation edit rate through optimizable edit costs. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 746–750. Association for Computational Linguistics, 2018. URL `https://www.aclweb.org/anthology/W18-6455`.

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

Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. Adversarial inference for multi-sentence video description. *CoRR*, abs/1812.05634, 2018. URL `http://arxiv.org/abs/1812.05634`.

Rebecca Passonneau. Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2006/pdf/636_pdf.pdf`.

Ramakanth Pasunuru and Mohit Bansal. Multi-task video captioning with video and entailment generation. *CoRR*, abs/1704.07489, 2017. URL `http://arxiv.org/abs/1704.07489`.

Tom Pelsmaeker and Wilker Aziz. Effective estimation of deep generative language models. *CoRR*, abs/1904.08194, 2019. URL `http://arxiv.org/abs/1904.08194`.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. Few-shot natural language generation for task-oriented dialog. February 2020. URL `https://www.microsoft.com/en-us/research/publication/few-shot-natural-language-generation-for-task-oriented-dialog/`.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Maja Popović. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL `https://www.aclweb.org/anthology/W15-3049`.

Maja Popovic, David Vilar, Eleftherios Avramidis, and Aljoscha Burchardt. Evaluation without references: Ibm1 scores as evaluation metrics. pp. 99–103, 07 2011.

Chris Quirk, Chris Brockett, and William Dolan. Monolingual machine translation for paraphrase generation. *EMNLP*, 2004.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2016.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. Plotmachines: Outline-conditioned generation with dynamic plot state tracking. In *arxiv*, 2020.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

Katharina Reinecke and Krzysztof Z. Gajos. Labinthewild: Conducting large-scale online experiments with uncompensated samples. In *CSCW '15*, 2015.

Ehud Reiter. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401, September 2018. URL `https://www.aclweb.org/anthology/J18-3002`.

Ehud Reiter. Ehud reiter's blog, 2019. URL `https://ehudreiter.com/blog-index/`.

Ehud Reiter and Anja Belz. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35:529–558, 2009a.

Ehud Reiter and Anja Belz. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Comput. Linguist.*, 35(4):529–558, December 2009b. URL `https://doi.org/10.1162/coli.2009.35.4.35405`.

Ehud Reiter and Robert Dale. Building applied natural language generation systems. *Cambridge University Press, Cambridge, UK*, 2000.

Ehud Reiter, Roma Robertson, and Liesl Osman. Lessons from a failure: Generating tailored smoking cessation letters. *Artif. Intell.*, 144:41–58, 2003.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016. URL `http://arxiv.org/abs/1602.04938`.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, 2018.

Brian Richards. Type/token ratios: what do they really tell us? *Journal of child language*, 14:201–9, 07 1987.

Alan Ritter, Colin Cherry, and Bill Dolan. Data-driven response generation in social media. In *Empirical Methods in Natural Language Processing (EMNLP)*, January 2011. URL `https://www.microsoft.com/en-us/research/publication/data-driven-response-generation-in-social-media/`.

M. Roemmele, A. Gordon, and R. Swanson. Evaluating story generation systems using automated linguistic analyses. *Workshop on Machine Learning for Creativity, at the 23rd SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2017)*, 2017.

Antti-veikko I. Rosti, Spyros Matsoukas, and Richard Schwartz. Improved word-level system combination for machine translation. *In Proc. of ACL 2007*, pp. 312–319, 2007.

Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *IEEE*, 1998.

Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D. Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language*, pp. 70–80, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W15-2812`.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. doi: 10.18653/v1/d19-1341. URL `http://dx.doi.org/10.18653/v1/d19-1341`.

William A. Scott. Reliability of content analysis: The case of nominal scale coding. *The Public Opinion Quarterly*, 19(3):321–325, 1955. ISSN 0033362X, 15375331. URL `http://www.jstor.org/stable/2746450`.

João Sedoc, Daphne Ippolito, Arun Kirubarajan, Jai Thirani, Lyle Ungar, and Chris Callison-Burch. Chateval: A tool for chatbot evaluation. 2019.

Abigail See, Peter J. Liu, and Christopher D. Manning. 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)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P17-1099`.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. Bleurt: Learning robust metrics for text generation, 2020.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. On accurate evaluation of gans for language generation. *CoRR*, abs/1806.04936, 2018. URL `http://arxiv.org/abs/1806.04936`.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. On accurate evaluation of GANs for language generation. *ICLR*, 2019. URL `https://openreview.net/forum?id=rJMcdsA5FX`.

Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C. Courville. Multiresolution recurrent neural networks: An application to dialogue response generation. *CoRR*, abs/1606.00776, 2016a. URL `http://arxiv.org/abs/1606.00776`.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. *CoRR*, abs/1605.06069, 2016b. URL `http://arxiv.org/abs/1605.06069`.

Naeha Sharif, Lyndon White, Mohammed Bennamoun, and Syed Afaq Ali Shah. Learning-based composite metrics for improved caption evaluation. In *Proceedings of ACL 2018, Student Research Workshop*, pp. 14–20, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P18-3003`.

Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, and Chandan K. Reddy. Neural abstractive text summarization with sequence-to-sequence models. *CoRR*, abs/1812.02303, 2018. URL `http://arxiv.org/abs/1812.02303`.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. RUSE: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 751–758, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W18-6456`.

Abhisek Singh and Wei Jin. Ranking summaries for informativeness and coherence withouth reference summaries. *Proceedings of the Twenty-Ninth International Florida Artifical Intelligent Research Society Conference*, 2016.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. pp. 223–231, 2006.

M. Sporleder, C.—Lapata. Discourse chunking and its application to sentence compression. *Proceedings of HLT/EMNLP*, pp. 257–264, 2005.

Peter Stanchev, Weiyue Wang, and Hermann Ney. EED: Extended edit distance measure for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 514–520, Florence, Italy, August 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-5359.

Manfred Stede and Carla Umbach. Dimlex: A lexicon of discourse markers for text generation and understanding. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98/COLING '98, pp. 1238–1242, USA, 1998. Association for Computational Linguistics.

Josef Steinberger and Karel Jezek. Evaluation measures for text summarization. *Computing and Informatics*, 28:251–275, 01 2009.

K. Steinberger, J.—Jezek. Sentence compression for the lsa-based summarizer. *Proceedings of the 7th International Conference on Information Systems Implementation and Modelling*, pp. 141–148, 2006.

Octavia-Maria Sulea. Recognizing textual entailment in Twitter using word embeddings. In *2nd Workshop on Evaluating Vector-Space Representations for NLP*, 2017.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 2.0: A continual pre-training framework for language understanding. *CoRR*, abs/1907.12412, 2019. URL http://arxiv.org/abs/1907.12412.

Manne Suneetha and Sheerin Fatima. Extraction based automatic text summarization system with hmm tagger. *International Journal of Soft Computing and Engineering*, ISSN: 2231-2307:2231–2307, 08 2011.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. URL http://arxiv.org/abs/1409.3215.

Liling Tan, Jon Dehdari, and Josef van Genabith. An awkward disparity between BLEU / RIBES scores and human judgements in machine translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pp. 74–81, Kyoto, Japan, October 2015. Workshop on Asian Translation. URL https://www.aclweb.org/anthology/W15-5009.

Rachel Tatman. Evaluating text output in nlp: Bleu at your own risk, 2019. URL https://towardsdatascience.com/evaluating-text-output-in-nlp-bleu-at-your-own-risk-e8609665a213.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Generating token-level explanations for natural language inference. *CoRR*, abs/1904.10717, 2019. URL http://arxiv.org/abs/1904.10717.

C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf. Accelerated dp based search for statistical translation. In *In European Conf. on Speech Communication and Technology*, pp. 2667–2670, 1997.

Jesús Tomás, Josep Àngel Mas, and Francisco Casacuberta. A quantitative method for machine translation evaluation. In *Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing: are evaluation methods, metrics and resources reusable?*, pp. 27–34, Columbus, Ohio, April 2003. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W03-2804.

A. M. Turing. Computing Machinary and Intelligence. *Mind*, LIX(236):433–460, 1950. URL https://doi.org/10.1093/mind/LIX.236.433.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, 2019. URL https://www.aclweb.org/anthology/W19-8643.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726, 2014. URL http://arxiv.org/abs/1411.5726.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014. URL `http://arxiv.org/abs/1411.4555`.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. 2015.

Nguyen Vo and Kyumin Lee. Learning from fact-checkers: Analysis and generation of fact-checking language. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pp. 335–344, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331248. URL `https://doi.org/10.1145/3331184.3331248`.

Alex Wang, Kyunghyun Cho, and Mike Lewis. Asking and answering questions to evaluate the factual consistency of summaries. 2020.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pp. 505–510, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W16-2342`.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. 2019.

Billy T. M. Wong and Chunyu Kit. Extending machine translation evaluation metrics with lexical cohesion to document level. pp. 1060–1068, 2019.

Lingfei Wu, Ian En-Hsu Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J. Witbrock. Word mover's embedding: From word2vec to document embedding. In *EMNLP*, 2018.

Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *ArXiv*, abs/2001.04063, 2020.

Qian Yang, Rebecca J. Passonneau, and Gerard de Melo. Peak: Pyramid evaluation via automated knowledge extraction. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016.

Lili Yao, Nanyun Peng, Ralph M. Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. *CoRR*, abs/1811.05701, 2018. URL `http://arxiv.org/abs/1811.05701`.

Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. Dependency-based discourse parser for single-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1834–1839, Doha, Qatar, October 2014. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D14-1196`.

R. Michael Young. Using grice's maxim of quantity to select the content of plan descriptions. *Artif. Intell.*, 115:215–256, 1999.

Hui Yu, Xiaofeng Wu, Jun Xie, Wenbin Jiang, Qun Liu, and Shouxun Lin. RED: A reference dependency based MT evaluation metric. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2042–2051, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C14-1193`.

Hui Yu, Xiaofeng Wu, Wenbin Jiang, Qun Liu, and ShouXun Lin. An automatic machine translation evaluation metric based on dependency parsing model. 08 2015.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 15–25. Association for Computational Linguistics, August 2017. URL `https://www.aclweb.org/anthology/W17-2603`.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In *NeurIPS*, 2019.

Rowan Zellers, Ari Holtzman, Elizabeth Clark, Lianhui Qin, Ali Farhadi, and Yejin Choi. Evaluating machines by their real-world language use. *ArXiv*, abs/2004.03607, 2020.

Jacob Zerr. Question generation using part of speech information. FINAL REPORT FOR REU PROGRAM AT UCCS, 2014. URL `http://cs.uccs.edu/~jkalita/work/reu/REU2014/FinalPapers/Zerr.pdf`.

Qiuyun Zhang, Bin Guo, Hao Wang, Yunji Liang, Shaoyang Hao, and Zhiwen Yu. Ai-powered text generation for harmonious human-machine interaction: Current state and future directions. *CoRR*, abs/1905.01984, 2019a. URL `http://arxiv.org/abs/1905.01984`.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SkeHuCVFDr`.

Ying Zhang, Stephan Vogel, and Alex Waibel. Interpreting bleu/nist scores: How much improvement do we need to have a better system. In *In Proceedings of Proceedings of Language Resources and Evaluation (LREC-2004*, pp. 2051–2054, 2004.

Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D. Manning, and Curtis P. Langlotz. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. *arXiv 1911.02541*, 2019b.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *EMNLP*, 2019.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93, 2020.

Wangchunshu Zhou and Ke Xu. Learning to compare for better training and evaluation of open domain natural language generation models, 2020.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. *CoRR*, abs/1802.01886, 2018. URL `http://arxiv.org/abs/1802.01886`.