RECEVAL: Evaluating Reasoning Chains via Correctness and Informativeness

Archiki Prasad **Swarnadeep Saha** **Xiang Zhou** **Mohit Bansal**
UNC Chapel Hill
{archiki, swarna, xzh, mbansal}@cs.unc.edu

Abstract

Multi-step reasoning ability is fundamental to many natural language tasks, yet it is unclear what constitutes a *good* reasoning chain and how to evaluate them. Most existing methods focus solely on whether the reasoning chain leads to the correct conclusion, but this answer-oriented view may confound the quality of reasoning with other spurious shortcuts to predict the answer. To bridge this gap, we evaluate reasoning chains by viewing them as *informal proofs* that derive the final answer. Specifically, we propose RECEVAL (**Re**asoning **C**hain **Eval**uation), a framework that evaluates reasoning chains through two key properties: (1) *correctness*, i.e., each step makes a valid inference based on the information contained within the step, preceding steps, and input context, and (2) *informativeness*, i.e., each step provides new information that is helpful towards deriving the generated answer. We implement RECEVAL using natural language inference models and information-theoretic measures. On multiple datasets, RECEVAL is highly effective in identifying different types of errors, resulting in notable improvements compared to prior methods. We demonstrate that our informativeness metric captures the expected flow of information in high-quality reasoning chains and we also analyze the impact of previous steps on evaluating correctness and informativeness. Finally, we show that selecting reasoning chains based on RECEVAL can improve downstream task performance.

Context: The moon is a kind of moon. Earth is a kind of planet. Moons orbit planets. Gravity causes orbits.
Question: What keeps the Moon orbiting Earth?
Model-generated Step-by-Step Rationales:
- Step 1: [Moon is a kind of moon] and [earth is a kind of planet], so [the moon and earth are planets].
- Step 2: [Gravity causes orbits], so [gravity causes moon to orbit earth].

Answer: Earth's gravity.

Figure 1: Model-generated step-by-step reasoning from Entailment Bank (Dalvi et al., 2021). Reasoning errors include: incorrect step inference (requires inferring *'moon orbits earth'*), and incorrect inference and uninformative (*'moon is a planet'* does not help answer the question). Reasoning Content Units (RCUs) are shown in parentheses ('[.]').

1 Introduction

The ability to reason is fundamental to many natural language processing tasks (Lin et al., 2019; Duan et al., 2020). A reasoning chain composes multiple reasoning steps together to accomplish an end task such as solving complex textual reasoning problems involving commonsense (Talmor et al., 2019; Huang et al., 2019; Aggarwal et al., 2021) and arithmetic (Hendrycks et al., 2021; Cobbe et al., 2021). Recent advances in scaling large language models have led to emergent reasoning capabilities, whereby a model is able to generate a reasoning chain in a few-shot manner (Wei et al., 2022; Chowdhery et al., 2022; Kojima et al., 2022). In most previous works, a model's reasoning capability is judged by its performance on the end task (Huang and Chang, 2022). This evaluation alone, however, is not ideal for understanding the reasoning ability of models, as it implies a narrow view of correctness based solely on the answer, and may confound the model's reasoning capabilities with unfaithful or spurious reasoning shortcuts leading to the correct answer (Creswell and Shanahan, 2022; Lyu et al., 2023). Thus, it is desirable to complement answer-oriented evaluation with an intrinsic evaluation of the quality of reasoning chains.

For a more comprehensive evaluation of reasoning chains, prior works leverage datasets containing human-written reasoning chains such as Entailment Bank (Dalvi et al., 2021), StrategyQA (Geva et al., 2021), etc., and develop supervised metrics that evaluate model-generated reasoning chains with respect to human-written ones (Clinciu et al., 2021;

Welleck et al., 2022; Saparov and He, 2023). However, this evaluation strategy may be infeasible due to the time-consuming and expensive nature of obtaining human-written (or gold) reasoning chains (Welleck et al., 2021; Tian et al., 2021; Han et al., 2022). Moreover, gold reference reasoning chains may not be unique, making the effectiveness of reference-based evaluations highly dependent on the selection and coverage of gold chains (Dalvi et al., 2021). A recent work, ROSCOE (Golovneva et al., 2023), took the first step towards reference-free evaluation by developing metrics based on generic reasoning errors like redundancy, hallucination, etc. In this work, we further explore this direction with the goal to formalize desired properties of reasoning chains and introduce additional metrics to assess these properties effectively.

In order to evaluate reasoning chains in a reference-free manner, it is important to ask what constitutes a *good* reasoning chain and what properties it should satisfy. We answer this question by viewing reasoning chains as *informal proofs* that lead to the final answer (Welleck et al., 2022; Jiang et al., 2023). While reasoning chains operate over natural language and may not adhere to the strict nature of formal proofs (Welleck et al., 2021), they serve a similar role in providing rationales for the final answer. Therefore, good reasoning chains share the same desirable properties as formal proofs. Conceptually, *each step in a reasoning chain should make a valid inference towards deriving the answer by leveraging prior information (i.e., previous steps or input context)*. In this work, we formalize this concept and propose a framework, RECEVAL (**Re**asoning **C**hain **Eval**uation) that defines good reasoning chains based on two properties: (1) *Correctness:* Each step generates a valid inference based on the information present within the step (*intra-step*) and past information derived in prior steps or available in the input context (*inter-step*); and (2) *Informativeness:* Each step provides new information that is helpful towards deriving the final answer (§3.2). Figure 1 contains an example where these properties are violated.

As part of our RECEVAL framework, we introduce a collection of reference-free metrics that measure the correctness and informativeness of reasoning chains (§4). To measure correctness, we decompose reasoning chains into fine-grained components called **R**easoning **C**ontent **U**nits (RCUs), each corresponding to a specific claim (§3.1, shown in Figure 1). We measure informativeness by computing the gain in information obtained by including each step in the reasoning chain towards deriving the final answer. We implement these metrics using a combination of $\mathcal{V}$-information (Xu et al., 2020; Ethayarajh et al., 2022; Hewitt et al., 2021), and Natural Language Inference models (Bowman et al., 2015; Williams et al., 2018).

To evaluate the strength of RECEVAL, we compare our metrics with multiple reference-free metrics (§6). Our meta-evaluation procedure is based on correlation with automatically perturbed and human-annotated errors in English reasoning chains from Entailment Bank (Dalvi et al., 2021) and GSM-8K (Cobbe et al., 2021) respectively. On Entailment Bank, we show that our metrics yield the highest correlation on 5 out of 6 error types, e.g., substantially improving correlation from $0.62 \rightarrow 0.86$ on hallucinations. Moreover, on the GSM-8K dataset, we observe that our metrics improve correlation from $0.28 \rightarrow 0.36$ on the overall quality measure and show superior performance in identifying 5 out of 7 error types. Next, we perform a comprehensive analysis of our metrics and demonstrate that RCUs facilitate evaluation of correctness of reasoning chains (§6.2) and that high-quality human-written reasoning chains typically exhibit a positive trend in information-gain (§6.3). Finally, we show that selecting high-scoring chains according to our proposed metrics also improves downstream task performance (§6.4).

In summary, our contributions are:

1. We propose RECEVAL, a framework for evaluating reasoning chains, building on top of the desired attributes of good reasoning chains: correctness and informativeness.
2. We propose various reference-free metrics to measure correctness and informativeness using NLI models and $\mathcal{V}$-information. They effectively identify various errors and substantially outperform prior methods in meta-evaluation.
3. We conduct a comprehensive study of our correctness and informativeness metrics and show that RECEVAL can also be used to improve the downstream performance of reasoning tasks.

2 Related Work

Typical evaluation metrics for text generation compute the similarity between two pieces of text using n-gram overlap (Papineni et al., 2002; Lin, 2004; Banerjee and Lavie, 2005) or model-based met-

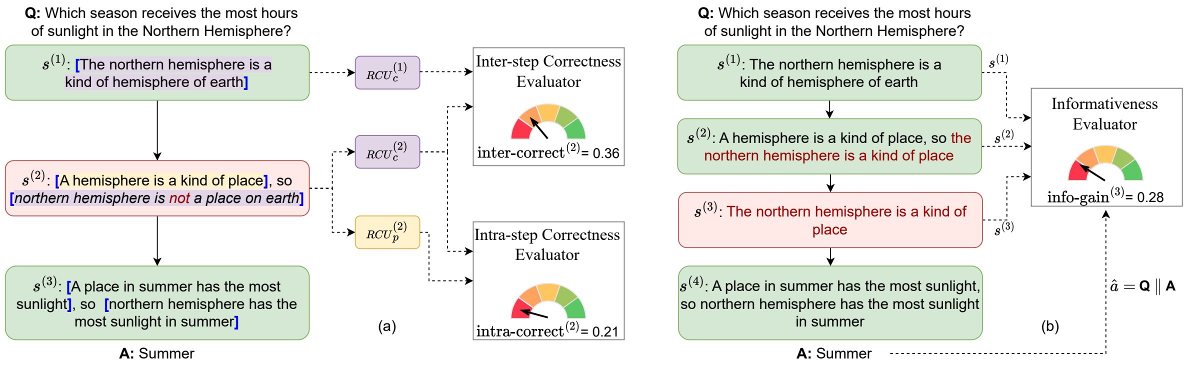

Figure 2: Illustration of evaluating a reasoning chain via ReCEval framework. In (a), we evaluate correctness of the second step using intra-correct$_{\text{entail}}$ and inter-correct metrics. For this, each step is split into reasoning content units (RCUs) indicated via '[.]' and categorized as premise-RCUs or conclusion-RCU. In (b), we evaluate informativeness of the third step towards the predicted answer given preceding steps via info-gain$_{\text{PVI}}$ (refer to §4).

rics based on embedding distance (Zhao et al., 2019; Zhang* et al., 2020; Sellam et al., 2020), information alignment (Deng et al., 2021), paraphrases (Thompson and Post, 2020), or pretrained text-generation models (Yuan et al., 2021; Fu et al., 2023a). While these metrics are relatively helpful for comparing machine-generated text to target text in summarization and machine translation tasks, they are not well-suited for evaluating reasoning chains that entail a coherent sequence of steps culminating in a final answer. Furthermore, since text generation metrics typically rely on references, they cannot be used to evaluate reasoning chains in a reference-free manner.

Some prior works on evaluating reasoning chains have proposed metrics that are based on specific construction and domain of the dataset, making them less generalizable. For instance, both FOLIO (Han et al., 2022) and PrOntoQA (Saparov and He, 2023) convert natural language reasoning chains to symbolic proofs using a fixed grammar and evaluate proofs in a reference-based manner. Similarly, Dalvi et al. (2021) compare model-generated reasoning trees to gold reasoning trees, but the latter may not be unique. Recently, Golovneva et al. (2023) propose RoscoE, a suite of metrics measuring semantic alignment, similarity, and logical inference, using a combination of reference-free and reference-based approaches. In our work, we focus on reference-free metrics that can be widely utilized to evaluate any reasoning chain. We first establish desired properties of good reasoning chains, such as correctness and informativeness, and then evaluate these properties using Reasoning Content Units (RCUs) and information-theoretic measures such as $\mathcal{V}$-information.

3 Reasoning Chains: Definition and Desired Properties

In this section, we formally define the concepts of reasoning chains and Reasoning Content Units (RCUs) in §3.1, and then describe the desiderata of good reasoning chains in §3.2.

3.1 Definitions

Reasoning Chain. Given a natural language reasoning task, let $\mathcal{X}$ denote the input context describing the reasoning problem. We define a reasoning chain $\mathcal{R} = \{s^{(1)}, \cdots, s^{(n)}\}$ as a multi-step rationale, consisting of n reasoning steps, used to arrive at a predicted answer $\hat{a}$. Reasoning chains can be human-written or model-generated (as in CoT prompting (Wei et al., 2022)).

Reasoning Content Unit (RCU). We further assume that each step $s^{(i)}$ is composed of one or many claims. We call these claims *Reasoning Content Units* (RCUs), as illustrated in Figure 2 via '[.]'. RCUs are conceptually similar and inspired by Summary Content Units (SCUs) used in fine-grained summary evaluation (Nenkova and Passonneau, 2004; Shapira et al., 2019; Zhang and Bansal, 2021). Visualizing a reasoning chain as a collection of steps and a step as a collection of RCUs facilitates fine-grained analysis and verification of a model's reasoning capabilities. The RCUs in a step $s^{(i)}$ typically can be split into a single conclusion-RCU, denoted by $\text{RCU}_c^{(i)}$ and remaining premise-RCUs, denoted by $\text{RCU}_{\boldsymbol{p}}^{(i)} = \{\text{RCU}_{p_j}^{(i)}\}_{j=1}^{t}$, where $t \geq 0$ is the number of premise-RCUs in the step. For example, in Figure 2(a), the final step $s^{(3)}$

consists of two RCUs of which the first ("a place ... most sunlight") is the premise and the second ("northern ... in summer") is the conclusion. We discuss how to identify RCUs in §4.5 and its utility to RECEVAL in §6.2.

3.2 Properties of Good Reasoning Chains

Reasoning chains are like informal proofs that lead to the final answer. Hence, we argue that the quality of reasoning steps in a reasoning chain should be measured based on two aspects: *correctness* and *informativeness*.

Correctness. First, every step in a reasoning chain should be correct. We say a step $s^{(i)}$ is correct if the corresponding conclusion $\text{RCU}_c^{(i)} \in s^{(i)}$ is also correct. The correctness of a step is governed by two factors: (1) *intra-step correctness* that evaluates whether the conclusion $\text{RCU}_c^{(i)}$ is correct, based on the information present in the premise units $\text{RCU}_{\boldsymbol{p}}^{(i)}$ within that step; and (2) *inter-step correctness* that evaluates whether $\text{RCU}_c^{(i)}$ is correct, given all the information present in the previous context (including input $\mathcal{X}$ and previous steps $s^{(<i)}$, represented by the corresponding RCUs $\{\text{RCU}_{\boldsymbol{p}}^{(<i)}, \text{RCU}_c^{(<i)}\}$). Intuitively, intra-step correctness evaluates the consistency of a claim within the step, while inter-step correctness complements it with a global check of consistency.[1] Figure 2(a) shows an example, in which $\text{RCU}_c^{(2)}$ in the second step $s^{(2)}$ does not follow from the premise $\text{RCU}_{\boldsymbol{p}}^{(2)}$ as it incorrectly concludes that the northern hemisphere is not a place on earth. This also contradicts the claim in the previous step $\text{RCU}_c^{(1)}$.

Informativeness. Besides correctness, we introduce another important property of a reasoning step: *informativeness*. The informativeness property complements correctness and aims to capture how helpful and important each reasoning step is for generating the final answer. Specifically, we observe that among multiple plausible inferences that can be made in a step, not all are equally relevant to answer the corresponding question. To capture such differences, we define informativeness as the extent to which a given step helps make progress in the "right direction" towards deriving the final answer. Figure 2(b) illustrates the role of informativeness. The third step $s^{(3)}$ does not contribute towards deriving the answer beyond the inference in the second step. While repetition or redundancy does not violate intra- or inter-step correctness, evaluating the reasoning chain based on informativeness allows us to identify such issues.

Next, we describe the technical details of our metrics that evaluate every reasoning step by itself (intra-step correctness), how it relates to the input and prior steps (inter-step correctness), and how it aids in solving the problem (informativeness).

4 RECEVAL: Evaluation of Reasoning Chains

With desired properties of reasoning chains introduced in §3, we now describe our evaluation framework, RECEVAL (**Re**asoning **C**hain **Eval**uation). RECEVAL builds upon concepts of textual entailment and $\mathcal{V}$-information. We first provide a brief background of $\mathcal{V}$-information in §4.1. Then, we introduce how RECEVAL captures intra-step correctness in §4.2, inter-step correctness in §4.3, and informativeness in §4.4. Finally, in §4.5, we present the complete framework combining step-level scores to evaluate the entire reasoning chain.

4.1 Background: $\mathcal{V}$-Information

Let X and Y denote two random variables. Their conditional entropy is defined as $H(Y|X) = \mathbb{E}[-\log P(Y|X)]$ (Shannon, 1948). However, computing it requires knowledge of the true joint distribution of X and Y which can be infeasible in practice. As an alternative, Xu et al. (2020) propose $\mathcal{V}$-conditional entropy using a model family $\mathcal{V}$ that learns to map from X to Y. It is defined as:

$$H_{\mathcal{V}}(Y|X) = \inf_{f\in\mathcal{V}} \mathbb{E}_{x,y\sim X,Y}(-\log f[x](y))$$

Each $f \in \mathcal{V}$ models the conditional distribution $P_f(Y|X)$. Thus, the model $\tilde{f} \in \mathcal{V}$, minimizing the above expectation, is optimized using a negative log-likelihood objective. Building on top of it, Xu et al. (2020) propose $\mathcal{V}$*-information* (also known as $\mathcal{V}$*-usable information*) which measures the amount of available information contained in X about Y that can be extracted using $\mathcal{V}$. It is defined as:

$$I_{\mathcal{V}}(X \to Y) = H_{\mathcal{V}}(Y|\varnothing) - H_{\mathcal{V}}(Y|X)$$

Here, we denote the models used to compute $H_{\mathcal{V}}(Y|X)$ and $H_{\mathcal{V}}(Y|\varnothing)$ (minimizing expecta-

[1] Note that it might so happen that 'correct' claims still lead to the incorrect answer (e.g., if the reasoning chain is incomplete). Evaluating whether a reasoning chain leads to the correct answer is thus complementary to our proposed intrinsic properties of good chains.

tion) as g and g' respectively.[2] Ethayarajh et al. (2022) propose *pointwise $\mathcal{V}$-information (*PVI*)* to measure the degree of usable information present in individual data points (x, y) as:

$$\text{PVI}(x \to y) = -\log g'[\varnothing](y) + \log g[x](y)$$

At a high level, we use PVI to extract the amount of information present within and across reasoning steps, as discussed in detail in §4.2 and §4.4.

4.2 Evaluation of Intra-Step Correctness

We propose two methods to measure the intra-step correctness of a reasoning step based on two complementary views of correctness. The first method exploits the connection between correctness and entailment, and the second method captures a more relaxed notion of correctness under the PVI framework. We describe the two methods below and compare their performance later in §6.2.

Entailment-based Intra-Step Correctness. Our first method aims to capture correctness by computing the entailment probability of the conclusion-RCU given the premise-RCUs within a step. Specifically, for a given step $s^{(i)}$, we evaluate if $\text{RCU}_{\boldsymbol{p}}^{(i)}$ entail $\text{RCU}_{c}^{(i)}$ (as introduced in §3):

$$\text{intra-correct}_{\text{entail}}^{(i)} = P_{\text{entail}}(\text{RCU}_{\boldsymbol{p}}^{(i)}; \text{RCU}_{c}^{(i)})$$

The premise-RCUs are concatenated and the entailment probability P_{entail} is computed using an off-the-shelf NLI model (Laurer et al., 2022). We enforce a *strict* definition of entailment, such that a conclusion-RCU that is neutral to the premise-RCUs also receives a low probability. We make this design choice because incorrect reasoning steps may contain hallucinations or non-factual claims that may not be supported by the premise-RCUs.

PVI-based Intra-Step Correctness. Our previous method requires premise-RCUs to *strictly* entail the conclusion-RCU. While this constraint should always stand in theory, reasoning steps in natural language can oftentimes be informal and still be perceived as correct when some of the premise-RCUs are omitted. To allow for such flexibility, here we create a relaxed criterion that evaluates the *ease* with which the conclusion can be drawn from the premise. Using the $\mathcal{V}$-information framework, we assess the ease of generating a conclusion-RCU based on the amount of useful information already contained in the premise-RCUs. Therefore, our metric can be written as:

$$\text{intra-correct}_{\text{PVI}}^{(i)} = \text{PVI}(\text{RCU}_{\boldsymbol{p}}^{(i)} \to \text{RCU}_{c}^{(i)})$$

This use of PVI is consistent with Padmakumar and He (2021), who use a pointwise information metric to evaluate the relevance of summary sentences.

4.3 Evaluation of Inter-Step Correctness

The aforementioned methods check for local correctness based on the premise-RCUs within a given step. Additionally, for a reasoning chain containing a large number of steps, we must also ensure that any new conclusion-RCU is also consistent with *all* known information either in the input $\mathcal{X}$ or in the conclusion-RCUs from prior steps. We measure this 'global' inter-step correctness by verifying the absence of contradiction between the current $\text{RCU}_{c}^{(i)}$ and prior information including $\mathcal{X}$ and all conclusion-RCUs so far $\text{RCU}_{c}^{(<i)}$. For instance, in step $s^{(2)}$ of Figure 2(a), we evaluate if $\text{RCU}_{c}^{(2)}$ is consistent with $\text{RCU}_{c}^{(1)}$. Similar to §4.2, we use the an NLI model to obtain the contradiction probability ($P_{contr.}$) and compute:

$$\text{inter-correct}^{(i)} = 1 - max_r(P_{\text{contr.}}(r; \text{RCU}_{c}^{(i)}))$$

where, $r \in \mathcal{X} \cup \{\text{RCU}_{c}^{(j)}\}_{j=1}^{i-1}$. We evaluate with respect to conclusion-RCUs only and exclude premise-RCUs from prior steps due to their overlap with input context $\mathcal{X}$. Empirically, we also do not observe any significant change in performance when premise-RCUs are excluded.

4.4 Evaluation of Informativeness in Reasoning Chains

As discussed in §3, in addition to correctness, each step in a good reasoning chain should also be informative of the final answer. We measure if adding a given step to the reasoning chain makes answering the question more likely using conditional PVI. This allows us to detect redundant steps that offer little benefit to predicting the final answer, (e.g., step $s^{(3)}$ in Figure 2(b)).

PVI-based Information Gain. To capture the contribution of a reasoning step, we measure the gain in information after adding it to the chain (constructed so far). A positive gain indicates that the step makes predicting the answer easier, whereas, a negative gain suggests otherwise. Inspired by Chen

[2]Consistent with established notation in $\mathcal{V}$-information work, $f[x](y)$ denotes $P_f(y|x)$ where f is a model. When $x = \varnothing$, we compute the probability of generating y directly.

Algorithm 1 Chain-level Scores in RECEVAL

Input: Context $\mathcal{X}$, Reasoning Chain $\mathcal{R}$, Predicted Answer $\hat{a}$
Output: Overall scores for $\mathcal{R}$ with each metric
for $s^{(i)} \in \mathcal{R}$ **do**
    $\texttt{RCU}_{\boldsymbol{p}}^{(i)}, \texttt{RCU}_{c}^{(i)} \leftarrow \texttt{content\_units}(s^{(i)})$
    $\text{score}_{\text{intra}}^{(i)} \leftarrow \text{intra-correct}^{(i)}(\texttt{RCU}_{\boldsymbol{p}}^{(i)}, \texttt{RCU}_{c}^{(i)})$
    $\text{score}_{\text{inter}}^{(i)} \leftarrow \text{inter-correct}^{(i)}(\texttt{RCU}_{c}^{(i)}, \mathcal{X}, s^{(<i)})$
    $\text{score}_{\text{info}}^{(i)} \leftarrow \text{info-gain}_{\text{PVI}}^{(i)}(s^{(\leq i)}, \hat{a})$
end for
$\text{score}_{\text{intra}} = \min_{i \in [1,n]}(\text{score}_{\text{intra}}^{(i)})$
$\text{score}_{\text{inter}} = \min_{i \in [1,n]}(\text{score}_{\text{inter}}^{(i)})$
$\text{score}_{\text{info}} = \min_{i \in [1,n]}(\text{score}_{\text{info}}^{(i)})$
return $\text{score}_{\text{intra}}, \text{score}_{\text{inter}}, \text{score}_{\text{info}}$

et al. (2022), who use conditional PVI relative to the question and gold answer, we compute the information provided by a step $s^{(i)}$ toward the predicted answer $\hat{a}$, conditioned on the previous steps $s^{(<i)}$. Based on §4.1, conditional PVI is defined as:

$$\text{PVI}(x \rightarrow y|z) = -\log g'[z](y) + \log g[z, x](y)$$

Specifically, we condition on previous steps ($z = s^{(<i)}$) to compute information present in a step ($x = s^{(i)}$) about the answer ($y = \hat{a}$), denoted as:

$$\text{info-gain}_{\text{PVI}}^{(i)} = \text{PVI}(s^{(i)} \rightarrow \hat{a}|s^{(<i)})$$

Unless mentioned otherwise, we use T5-large (Raffel et al., 2020) as our model family $\mathcal{V}$. In §6.3, we analyze our informativeness metric in detail and discuss alternate implementations of $\text{info-gain}_{\text{PVI}}$.

4.5 RECEVAL: Overall Algorithm

We now describe our overall RECEVAL algorithm for evaluating reasoning chains based on the aforementioned step-level metrics.

Identifying RCUs. We first split each step into its constituent RCUs by using an off-the-shelf Semantic Role Labeling (SRL) model that decomposes a sentence into semantic triplets containing 'subject-verb-object' frames (Shi and Lin, 2019; Zhang and Bansal, 2021). This yields multiple frames for each sentence, from which we extract maximal non-overlapping frames and treat them as our units. The extracted RCUs within each step are then classified as either premise or conclusion RCUs based on its location within the sentence and the sentence structure (refer to Appendix A).

Overall Reasoning Chain Evaluation. Once we decompose a step into RCUs, we evaluate its correctness and informativeness using the metrics described in §4. Next, we combine the step-level evaluations in order to determine the overall quality of a reasoning chain. Following the scoring setup in Golovneva et al. (2023), we consider a reasoning chain is only as good as its least correct or least informative step. Therefore, given a reasoning chain and an evaluation metric, we aggregate step-level scores using a '$\min$' operation. This is outlined in Algorithm 1. These chain-level scores for each metric can then be used to identify various types of errors (discussed in §6).

Additional implementation details of RECEVAL including model checkpoints, identifying RCUs, and computing PVI are present in Appendix A.

5 Meta-Evaluation Setup

We evaluate the effectiveness of a metric in detecting errors in reasoning chains using the meta-evaluation framework established by Golovneva et al. (2023). Specifically, we compute the correlation between the metric and ground-truth annotations that indicate the presence of particular types of errors. In this section, we first describe the datasets containing ground-truth error annotations in reasoning chains (in §5.1), followed by the baseline evaluation metrics that we compare RECEVAL to (in §5.2). Lastly, we introduce the correlation measure for meta-evaluation (in §5.3).

5.1 Meta-Evaluation: Datasets

We test RECEVAL on two complex English reasoning datasets, Entailment Bank (Dalvi et al., 2021) and GSM-8K (Cobbe et al., 2021), which contain high-quality multi-step reasoning chains covering diverse types of reasoning. Additional details and examples can be found in Appendix B.

Entailment Bank. Entailment Bank (EB) is a deductive reasoning dataset in which the goal is to derive a hypothesis based on an initial set of facts and it consists of human-written reasoning chains. Golovneva et al. (2023) emulate reasoning errors on EB via programmatic perturbations, which henceforth, will be referred to as EB-regular. While we use the same error types, we also observe that the content in a gold reasoning chain overlaps with the input context. This makes perturbations (in EB-regular) applied to overlapping facts in $\mathcal{R}$ relatively easy to measure and thus, may lead to higher correlations (examples in Appendix B). Hence, in order to construct a more realistic and challenging dataset, we apply perturbations only to intermediate inferences *not* included in the context. We refer to our set of perturbed reasoning chains as EB-challenge (contains a validation and test split). Across both EB-regular and EB-challenge, the categories of errors include hallucinations (HALL), negation (NEG), and swap (SWAP), which involve replacing the intermediate inference with a distractor, negating the inference, and interchanging steps respectively. We also include variations of informativeness errors such as verbatim repetition (REP), adding a paraphrase of an inference (PAR), or a sentence irrelevant to the reasoning problem (RED).

GSM-8K. GSM-8K contains grade school math word problems requiring mathematical reasoning. We directly use human judgments collected by Golovneva et al. (2023) evaluating model-generated CoT steps provided by Wei et al. (2022). The dataset contains two overall scores measuring the quality (QUAL) and coherence (COH) of the reasoning chain on a Likert scale (1-5). Furthermore, the annotations contain binary responses corresponding to the presence of specific errors in each step. These include factuality issues (FACT), logical deduction errors (LOGIC), hallucinations (HALL), presence of redundant or irrelevant information (RED), unnecessary paraphrasing (REP), commonsense errors (COM), and arithmetic errors (MATH). A reasoning chain is said to contain these errors if any of its steps contains this type of error. We refer readers to Golovneva et al. (2023) for a detailed description of the data collection process.

Note that in GSM-8K, the same reasoning chain can have multiple errors, whereas, in EB, we consider one error at a time. For a summary of errors in both datasets, refer to Table 16 (Appendix B).

5.2 Meta-Evaluation: Baselines

We now describe the baseline metrics with which we compare our RECEVAL metrics. Following Golovneva et al. (2023), we choose a variety of text-generation metrics measuring n-gram match such as ROUGE-1, ROUGE-2 and ROUGE-L (Lin, 2004), and pretrained model-based metrics including PRISM (Thompson and Post, 2020), BERTScore (Zhang* et al., 2020), BARTScore (Yuan et al., 2021), CTC Relevancy and Consistency (Deng et al., 2021). Each metric compares the reasoning chain $\mathcal{R}$ (as a paragraph) with the input context $\mathcal{X}$. We also compare against semantic similarity (SS), semantic alignment (SA), and logical inference (LI) metrics from ROSCOE. For ROSCOE-SA, and -SS, we use the text-similarity models finetuned on reasoning chains (Golovneva et al., 2023). Additionally, we group the reference-free metrics from ROSCOE that measure redundancy ('repetition-token' and 'repetition-step') as ROSCOE-REP. This enables a direct comparison with ROSCOE on two desired properties: correctness and informativeness. To evaluate correctness, we compare with ROSCOE-SA, -SS, and -LI, while for informativeness, we compare with ROSCOE-SA, -SS, and -REP.

5.3 Meta-Evaluation: Correlation Measure

After scoring each reasoning chain with either RECEVAL (from §4) or baseline metrics (from §5.2), we assess whether the score can indicate the presence or absence of each type of error. Identical to Golovneva et al. (2023), we use the Somer's-D correlation (Somers, 1962), which measures ordinal association between two dependent quantities. In our case, we evaluate a metric S against the random variable indicating whether the chain is erroneous ($E \in \{0, 1\}$). Using Kendall's τ coefficient, Somer's-D correlation is defined as:

$$D_{SE} = \tau(E, S)/\tau(E, E).$$

Consistent with Golovneva et al. (2023), when multiple metrics are available (as in ROSCOE or RECEVAL), we compute the correlation with each variant and report the highest obtained correlation.

6 Results and Discussion

In this section, we demonstrate that RECEVAL is effective at identifying various types of errors (in §6.1), followed by a comprehensive analysis of correctness and informativeness metrics in §6.2 and §6.3 respectively. Finally, in §6.4, we discuss the downstream utility of RECEVAL in improving performance on reasoning tasks.

6.1 Effectiveness of RECEVAL

In this section, we present our main meta-evaluation results and show that RECEVAL can effectively identify different types of errors in Entailment Bank and GSM-8K datasets.

Entailment Bank. Among the error types in EB-challenge (described in §5.1), hallucination, nega-

Metric	Error Types		
	HALL	NEG	SWAP
ROUGE-1	0.01	0.02	0.13
ROUGE-2	-0.01	-0.02	0.14
ROUGE-L	-0.04	0.01	0.10
BERTScore	0.09	0.02	0.07
BARTScore	0.00	-0.01	0.07
PRISM	0.27	0.03	0.08
CTC Relevancy	0.09	-0.04	-0.05
CTC Consistency	0.00	-0.05	-0.03
ROSCOE-SA	<u>0.62</u>	0.40	<u>0.22</u>
ROSCOE-SS	0.34	0.40	0.09
ROSCOE-LI	0.20	<u>0.82</u>	0.16
RECEVAL-correctness	**0.86**	**0.89**	**0.34**

(a) Correctness

Metric	Error Types		
	REP	PAR	RED
ROUGE-1	0.45	0.26	0.15
ROUGE-2	0.43	0.21	0.11
ROUGE-L	0.08	0.09	0.10
BERTScore	0.24	0.16	0.12
BARTScore	0.11	0.12	0.08
PRISM	0.15	0.11	0.09
CTC Relevancy	0.24	0.14	0.10
CTC Consistency	0.25	0.15	0.12
ROSCOE-SA	**0.83**	<u>0.64</u>	0.51
ROSCOE-SS	0.81	0.62	<u>0.54</u>
ROSCOE-REP	**0.83**	<u>0.64</u>	0.48
RECEVAL-informativeness	<u>0.66</u>	**0.68**	**0.67**

(b) Informativeness

Table 1: Comparison of Somer's D correlation scores using standard text-generation metrics, ROSCOE, and RECEVAL on EB-challenge (test split). Table 12 in Appendix C shows similar trends on EB-regular. The highest correlation is highlighted in bold and the second-highest correlation is underlined (higher correlation is better).

Metric	Error Types								
	QUAL	COH	COM	FACT	HALL	RED	REP	LOGIC	MATH
ROUGE-1	0.12	0.20	0.07	0.16	0.27	0.04	0.22	0.07	0.23
ROUGE-2	0.09	0.14	0.06	0.10	0.17	-0.02	0.56	0.03	0.11
ROUGE-L	0.17	0.27	0.19	0.17	0.18	0.05	0.56	0.12	0.21
BERTScore	0.19	0.23	0.12	0.13	0.20	0.13	0.94	0.15	0.13
BARTScore	0.01	0.03	-0.05	0.04	-0.25	-0.26	0.42	0.00	-0.55
PRISM	-0.11	-0.07	-0.10	-0.04	-0.39	-0.46	-0.09	-0.17	-0.34
CTC Relevancy	-0.09	-0.15	-0.08	-0.11	0.01	-0.37	0.57	-0.11	-0.09
CTC Consistency	-0.16	-0.20	-0.21	-0.13	-0.01	-0.32	0.56	-0.17	-0.02
ROSCOE-SA	0.20	0.19	0.19	0.08	0.22	0.39	0.79	0.18	**0.44**
ROSCOE-SS	0.20	0.17	0.17	0.14	0.25	<u>0.51</u>	<u>0.87</u>	0.15	0.23
ROSCOE-LI	0.28	0.26	0.18	<u>0.34</u>	0.22	0.35	**0.98**	<u>0.22</u>	0.09
ROSCOE-REP	0.20	0.19	0.19	0.14	0.25	<u>0.51</u>	<u>0.87</u>	0.18	**0.44**
RECEVAL-correctness	**0.36**	**0.31**	**0.21**	**0.37**	**0.28**	0.40	0.63	**0.25**	0.24
RECEVAL-informativeness	<u>0.30</u>	<u>0.29</u>	<u>0.19</u>	0.26	<u>0.26</u>	**0.55**	<u>0.87</u>	0.21	<u>0.32</u>

Table 2: Comparison of Somer's D correlation scores using standard text-generation metrics, ROSCOE, and RECEVAL on GSM-8K (test split) containing human-annotated errors from Golovneva et al. (2023).

tion, and swap errors are representative of correctness issues while repetition, paraphrase, and redundancy errors point to uninformativeness. We present the meta-evaluation results in Table 1. First, in Table 1a, we observe that the correctness metrics in RECEVAL outperform all previous baseline metrics. Compared to standard text-generation metrics, RECEVAL-correctness substantially improves correlation from $0.27 \rightarrow 0.86$ and $0.14 \rightarrow 0.34$ on hallucination and swap errors respectively. Notably, we also obtain higher correlations from $0.62 \rightarrow 0.86$, $0.82 \rightarrow 0.89$, and $0.22 \rightarrow 0.34$ as compared to ROSCOE on hallucinations, negation, and swap errors respectively. Next, in Table 1b, we evaluate the ability to identify reasoning chains with uninformative steps in the form of repetition, paraphrasing, and redundancy (refer to Table 16 for overview). The informativeness metric in RECEVAL outperforms all baselines on complex errors like paraphrasing and redundancy errors by at least $0.64 \rightarrow 0.68$ and $0.54 \rightarrow 0.67$ respectively compared to ROSCOE. For verbatim repetition (REP), our metric obtains higher correlation than all standard text-generation metrics ($0.45 \rightarrow 0.66$), however, ROSCOE achieves the best performance using metrics based on sentence similarity. In Table 12 in Appendix C, we show that similar trends hold when comparing metrics on EB-regular.

Method	intra-correct			inter-correct		
	HALL	NEG	SWAP	HALL	NEG	SWAP
w/o RCUs	-	-	-	0.12	0.83	0.11
our RCUs	0.71	0.84	0.37	0.14	0.90	0.16
gold RCUs	0.89	0.94	0.54	0.16	0.96	0.16

Table 3: Comparison of correctness metrics in RECEVAL on EB-challenge (validation split) with different RCU selection. Specifically, we use $\text{intra-correct}_{\text{entail}}$.

Method	Error Types		
	HALL	NEG	SWAP
$\text{intra-correct}_{\text{entail}}$	0.71	0.84	0.37
$\text{intra-correct}_{\text{PVI}}$	0.86	0.16	0.38
$\text{intra-correct}_{\text{no-contr.}}$	0.02	0.82	0.08

Table 4: Comparison of different implementations of the intra-step correctness metric on the validation split of EB-challenge.

GSM-8K. In Table 2, we present the meta-evaluation results with respect to human judgments on model-generated reasoning chains from GSM-8K. We observe that our RECEVAL framework outperforms all the baseline metrics on a majority of error types. Compared to the text-generation metrics, we obtain higher correlations across all errors. Additionally, on the overall quality (QUAL) and coherence (COH) measures, we obtain higher correlation than ROSCOE-LI and ROSCOE semantic metrics, by up to $0.28 \rightarrow 0.36$ and $0.20 \rightarrow 0.36$ respectively. Moreover, our metrics show higher correlation on commonsense (COM), factuality (FACT), hallucination (HALL), and logical (LOGIC) errors by up to 0.06. For errors focusing on informativeness, we observe that our informativeness metric yields the highest correlation on redundancy ($0.51 \rightarrow 0.55$) and comparable correlations on repetition errors.[3] Overall, we find that metrics in RECEVAL obtain the highest and second highest correlations for most errors. Our metrics are not specifically designed to handle arithmetic errors. One way to do so is to use external calculators or ROSCOE-REP, which demonstrates a better correlation for math errors (using sentence similarity models) but we leave this exploration for future work.

6.2 Analysis of Correctness Metrics

We further analyze intra- and inter-step correctness metrics using the EB dataset by asking the following research questions.

How do the design choices of RCUs affect correctness evaluation? Our correctness measures rely on automatically identified RCUs that are not typically annotated in a reasoning chain (§4). To understand the role played by RCUs in measuring correctness, we compare different variants of our metric implemented with (i) identified RCUs, (ii) no RCUs (by treating a step as a whole), and (iii) gold RCU annotations (i.e., oracle setting). We extract gold RCUs for each step using the reasoning trees supplied in the EB dataset (details in Appendix D). We present the results in Table 3. Overall, we show that RCU decomposition is crucial to RECEVAL, especially because it enables measuring intra-step correctness, which in turn leads to more accurate identification of hallucinations and swap errors. Additionally, we observe that gold RCUs improve both correctness metrics and yield higher correlation across errors (up to 0.20). Nevertheless, our identified RCUs strengthen correctness evaluation and future work can further bridge the gap between the two settings.

How do different methods (entailment vs. PVI) impact the correctness metric? As described in §4.2, correctness can be measured using various viewpoints (e.g., based on entailment or PVI). Using intra-step correctness, we study how these implementations affect the overall correlation for different error types. We compare our two proposed methods: $\text{intra-correct}_{\text{entail}}$ and $\text{intra-correct}_{\text{pvi}}$. Additionally, we also view correctness based on the absence of contradictions between premise and conclusion RCUs, denoted as $\text{intra-correct}_{\text{no-contr.}}$. Table 4 provides a comparison of the above three intra-correctness methods on different error types. We observe that the PVI method is better at identifying hallucinations while the entailment method works better for negation errors. Both scores perform comparably on the swap errors. Furthermore, $\text{intra-correct}_{\text{no-contr.}}$ can only capture negation, but it still underperforms $\text{intra-correct}_{\text{entail}}$. Thus, we conclude that $\text{intra-correct}_{\text{entail}}$ and $\text{intra-correct}_{\text{PVI}}$ have different degrees of effectiveness depending on the type of error and can be used in a complementary way. We conduct a similar study for inter-step correctness in Appendix D.

How does the amount of previous information impact inter-step correctness? In inter-step cor-

[3]The relative frequency of REP errors is very low, therefore, label imbalance results in spurious correlation between REP and overall coherency COH when using ROSCOE-LI.

Method	Error Types		
	HALL	NEG	SWAP
inter-correct ($k = 1$)	0.08	0.79	0.14
inter-correct ($k = 2$)	0.10	0.84	0.17
inter-correct ($k = all$)	0.14	0.90	0.16

Table 5: Comparison of inter-correct metric with different amounts of prior information on validation split of EB-challenge. The number of preceding steps used to compute scores is denoted by k.

Step Granularity	Error Types		
	HALL	NEG	SWAP
Step = RCU	0.46	0.87	0.28
Step = sentence	0.86	0.90	0.38
Step = $\mathcal{R}$	0.17	0.32	0.13

Table 6: Comparing performance of correctness metrics in RECEVAL for different step boundaries on validation split of EB-challenge.

rectness (§4.3), we evaluate if a given step violates any of the conclusion-RCUs from previous steps or input context $\mathcal{X}$. We now investigate how the amount of prior information impacts the performance of inter-step correctness by checking contradictions with respect to only k preceding steps. We explore three variants using $k = 1, 2$, and all in Table 5. We observe that using only immediately preceding steps (i.e., $k = 1, 2$) results in an up to 0.11 decrease in correlation for hallucination and negate errors. Hence, it is important to evaluate inter-step correctness with respect to all previous steps to best identify possible errors.

What constitutes a step and how does its granularity impact RECEVAL's effectiveness? Unlike formal proofs, it is not straightforward to demarcate the step boundaries in natural language reasoning chains, which can in turn affect high-quality evaluation of reasoning. To show this (Table 6), we compare our setting (where each step is a sentence) with two variants where the step size is either too small or too big. In particular, we compare the performance of our correctness metric when (i) each RCU is a step, (ii) each sentence is a step, and (iii) the entire reasoning chain is the one and only step. As expected, we observe that both extremes of step boundaries lead to a decrease in correlation across error types. Using RCU-level step boundaries leads to lower correlations on hallucination and swap errors. However, treating the entire reasoning chain as a single step results in significantly lower correlations on all errors. This happens because the resulting step includes multiple intermediate conclusion-RCUs, of which only the last conclusion is evaluated. This highlights the importance of choosing appropriate step boundaries when evaluating multi-step rationales and considering each sentence as a step works well in practice.

Reasoning Chain	AMI_k		
	$k = 1$	$k = 2$	$k = 3$
Uninformative (REP)	36.4	69.4	80.7
Uninformative (PAR)	35.3	70.5	81.4
Uninformative (RED)	38.6	73.4	82.8
Gold	72.7	87.7	92.0

Table 7: Fraction (%) of API_k chains (gold or uninfromative) in validation split of EB-challenge.

6.3 Analysis of Informativeness Metric

In this section, we explore how informativeness varies across steps within a reasoning chain and compare alternative implementations of info-gain.

How does informativeness vary across steps? Ideally, if each step in a good reasoning chain adds useful information about solving the underlying problem, then information gain should be positive for all steps in a reasoning chain. We ask if such a property holds for human-written reasoning chains and if so, how they compare to uninformative chains (i.e., chains with any uninformative step). To quantify this trend, we define a property called *Approximately Positive Information-gain* (API) across steps (additional details in Appendix E). A reasoning chain is considered to be API_k across steps, if all k contiguous steps are collectively more informative than the preceding steps (i.e., they yield a positive information gain). Using our $\text{info-gain}_{\text{PVI}}$ metric, we define API_k as:

$$\mathrm{API}_k(\mathcal{R}) = \begin{cases} 1 & \sum_{j=i}^{i+k-1} \text{info-gain}^{(j)}_{\text{PVI}} > 0, \forall s^{(i)} \in \mathcal{R} \\ 0, & \text{otherwise.} \end{cases}$$

If $k = 1$, this is equivalent to strictly positive information-gain for all steps, i.e., if each step helps derive answer $\hat{a}$. Table 7 shows the relative frequency of API_k gold and uninformative chains. We find that 72% of gold reasoning chains have positive information-gain for all steps ($\mathrm{API}_1 = 1$) which is considerably higher than uninformative

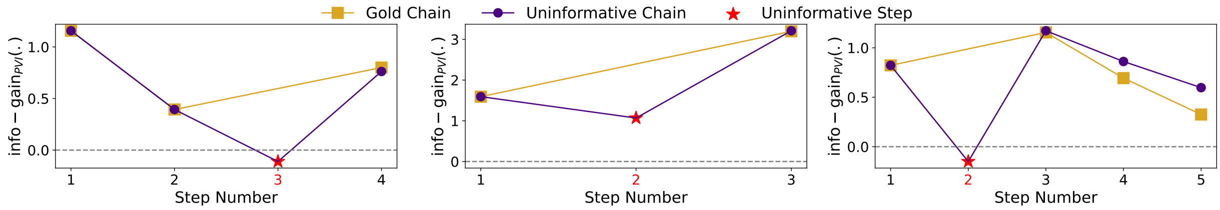

Figure 3: Trends in information gain of steps across gold and uninformative (REP) reasoning chains from EB-challenge. The position of the added uninformative step is highlighted in red on the x-axis and via '⋆' marker.

Method	Error Types		
	REP	PAR	RED
info-gain$_{\text{PVI}}$	0.67	0.66	0.65
info-gain$_{\text{LL}}$	0.58	0.60	0.60

Table 8: Comparison of informativeness metric computed using probabilities from trained PVI models and pretrained LMs on validation split of EB-challenge.

Method	Error Types		
	REP	PAR	RED
info-gain$_{\text{PVI}}$ $(k = 1)$	0.65	0.66	0.64
info-gain$_{\text{PVI}}$ $(k = 2)$	0.70	0.69	0.68
info-gain$_{\text{PVI}}$ $(k = all)$	0.65	0.64	0.63

Table 9: Comparison of informativeness metric of RECEVAL on validation split of EB-challenge using different amounts prior steps (k) in the reasoning chain.

chains (35-38%). Furthermore, we observe that 87% of gold reasoning chains have positive gains when considering two consecutive steps, and as high as 92% when considering three steps at a time. In Figure 3, we show the informativeness trends in three randomly chosen reasoning chains. As expected, adding an uninformative step results in a negative (or minimally positive) information-gain, thereby indicating that the uninformative step is least helpful in deriving the answer. Thus, we conclude that most steps in high-quality reasoning chains have positive information gain. Moreover, we demonstrate that info-gain$_{\text{PVI}}$ can fairly capture this trend and can distinguish between informative and uninformative reasoning chains.

How does the underlying probability model affect info-gain? In §4.4, we use conditional PVI to measure the information-gain which requires fine-tuning models to learn the distribution of text in reasoning steps. If training data in the form of gold reasoning steps is not available, we propose an alternative that uses the log-likelihood of generating text in the step from a pretrained LM, called info-gain$_{\text{LL}}$. To this end, we use GPT-2 XL (Radford et al., 2019), an auto-regressive LM, to calculate the log-likelihoods and as the model family used in computing PVI.[4] We provide a comparison of both methods in Table 8 and observe that info-gain$_{\text{PVI}}$ yields higher correlation (difference of at least 0.05) as compared to info-gain$_{\text{LL}}$ across all error types. In summary, while using a fine-tuned LM can identify errors more effectively, off-the-shelf LMs can also achieve decent performance when gold chains are not accessible for fine-tuning.

How does info-gain vary based on the number of preceding steps? Finally, we are interested in analyzing the effect of the number of past steps conditioned on for computing info-gain. Instead of measuring the gain relative to all the preceding reasoning steps, we also consider using only k preceding steps to compute information gain. In Table 9, we find that using $k = 2$ prior steps outperforms $k = 1$ consistently with nearly 0.04 higher correlation across error types. However, using all prior steps is comparable to $k = 1$ step. We suspect that the distinction between informative and uninformative chains becomes more pronounced when the reasoning chain is truncated and some of the required information for reasoning is absent from the context. Thus, we use $k = 2$ to compute info-gain in our final experiments in §6.1.

6.4 Downstream Utility of RECEVAL Metrics

While evaluating reasoning chains in itself is a challenging yet important task, we also explore if our proposed metrics can be applied to increase downstream task performance. Specifically, we study whether higher-quality reasoning chains (ranked based on our metrics) leads to more accurate an-

[4]We use GPT-2 XL instead of T5-large as the latter is not an auto-regressive LM and cannot reliably be used to estimate log-likelihood without finetuning.

Method	Accuracy (%)
Greedy Decoding	17.3
Sampling + ROSCOE (LI)	19.0
Sampling + ROSCOE (SA, SS)	17.8
Sampling + ROSCOE (REP)	18.6
Sampling + RECEVAL (correctness)	19.6
Sampling + RECEVAL (informativeness)	18.7
Sampling + RECEVAL (both)	20.5

Table 10: Applying RECEVAL to improve downstream task performance on GSM-8K using FLAN T5-XXL.

swers when using CoT prompting.

Experimental Setup. We generate reasoning chains for the GSM-8K dataset using the FLAN T5-XXL model (Chung et al., 2022) with the instruction: "Answer the following question by reasoning step-by-step". During decoding, we sample 20 reasoning chains, select the best one based on ROSCOE or RECEVAL metrics and compare them to chains obtained via greedy decoding. Since both ROSCOE and RECEVAL contain multiple metrics, we use a simple aggregation strategy for selecting reasoning chains. We select the chain with the highest scores on all metrics wherever possible. If such a chain does not exist, we rank chains based on each metric and select the chain with the lowest cumulative rank. Among RECEVAL metrics, we consider correctness, informativeness metrics separately as well as their combination. Similarly, for ROSCOE, we consider three settings using: (i) ROSCOE-LI (which yielded the best performance on overall QUAL and COH measures in Table 2), (ii) ROSCOE-REP (analogous to informativeness), and (iii) non-repetition metrics from ROSCOE-SA and ROSCOE-SS (analogous to correctness).[5]

Results. We present the results in Table 10. We observe that RECEVAL improves the QA accuracy by 3.2 points over standard greedy decoding, when filtering based on both correctness and informativeness. Using only correctness and informativeness yields an improvement of 2.3 and 1.4 points respectively. In comparison, different combinations of ROSCOE metrics improve accuracy by up to 1.7 points. This is an initial promising demonstration of the downstream utility of reasoning chain evaluation metrics. Future work can explore methods of combining these metrics with other sampling strategies (Wang et al., 2023; Fu et al., 2023b) for improving the reasoning capability of LLMs.

[5]We observed that combining all ROSCOE metrics did not further improve accuracy. Future works can explore more complex aggregation strategies for combining multiple metrics in ROSCOE and RECEVAL.

7 Conclusion and Future Work

We presented RECEVAL, a framework for evaluating reasoning chains based on correctness and informativeness. We proposed reference-free metrics for measuring these properties that are based on entailment and pointwise $\mathcal{V}$-information, leveraging granular claims in reasoning chains called Reasoning Content Units (RCUs). Our method considerably outperforms previous baseline metrics, as shown by meta-evaluation on multiple datasets. We also perform detailed analysis of our metrics and demonstrate that RECEVAL is effective in various settings, and leads to downstream improvement in task performance.

An interesting assumption for future work to address is that all knowledge typically needed to evaluate the correctness of a reasoning step is explicitly present as part of the input or the intermediate reasoning steps. In scenarios where correctness depends on implicit knowledge, we rely on the choice of underlying models (described in Appendix A) which are built on top of pre-trained LMs and are known to capture a lot of background knowledge (Petroni et al., 2019; Roberts et al., 2020). However, inferences that rely on substantial implicit knowledge may not be best evaluated through current metrics. While current evaluation frameworks focus on evaluating the quality of model-generated reasoning chains, Wei et al. (2022) note that the chain itself may not faithfully reflect the internal reasoning process of the model. This remains an open question for future work to address.

Acknowledgements

We thank Peter Hase, Prateek Yadav, and Shiyue Zhang for their helpful comments. This work was supported by NSF-CAREER Award 1846185, NSF-AI Engage Institute DRL-2112635, DARPA Machine Commonsense (MCS) Grant N66001-19-2-4031, and a Google Ph.D. Fellowship. The views contained in this article are those of the authors and not of the funding agency.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and

Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065, Online. Association for Computational Linguistics.

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

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hanjie Chen, Faeze Brahman, Xiang Ren, Yangfeng Ji, Yejin Choi, and Swabha Swayamdipta. 2022. Rev: Information-theoretic evaluation of free-text rationales. *arXiv preprint arXiv:2210.04982*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Miruna-Adriana Clinciu, Arash Eshghi, and Helen Hastie. 2021. A study of automatic metrics for the evaluation of natural language explanations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2376–2387, Online. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7580–7605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Nan Duan, Duyu Tang, and Ming Zhou. 2020. Machine reasoning: Technology, dilemma and future. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 1–6, Online. Association for Computational Linguistics.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding Dataset Difficulty with $\mathcal{V}$-Usable Information. In *International Conference on Machine Learning*, pages 5988–6008. PMLR.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023a. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023b. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, et al. 2022. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *Advances in Neural Information Processing Systems*.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. 2021. Conditional probing: measuring usable information beyond a baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1626–1639, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Albert Qiaochu Jiang, Sean Welleck, Jin Peng Zhou, Timothee Lacroix, Jiacheng Liu, Wenda Li, Mateja Jamnik, Guillaume Lample, and Yuhuai Wu. 2023. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. In *The Eleventh International Conference on Learning Representations*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Moritz Laurer, W v Atteveldt, Andreu Casas, and Kasper Welbers. 2022. Less annotating, more classifying–addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli.

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

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379*.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Vishakh Padmakumar and He He. 2021. Unsupervised extractive summarization using pointwise mutual information. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2505–2512, Online. Association for Computational Linguistics.

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

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5418–5426, Online. Association for Computational Linguistics.

Abulhair Saparov and He He. 2023. Language models can (kind of) reason: A systematic formal analysis of chain-of-thought. In *International Conference on Learning Representations*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 682–687, Minneapolis, Minnesota. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Robert H Somers. 1962. A new asymmetric measure of association for ordinal variables. *American sociological review*, pages 799–811.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. Naturalproofs: Mathematical theorem proving in natural language. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. 2022. Naturalprover: Grounded mathematical proof generation with language models. In *Advances in Neural Information Processing Systems*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. 2020. A theory of usable information under computational constraints. *International Conference on Learning Representations*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Shiyue Zhang and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6617–6632, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

A ReCEval: Implementation Details

In this section, we describe additional implementation details of our ReCEval framework.

Use of External Tools. We use three categories of models: (i) Semantic Role Labeling (SRL) models for identifying RCUs, (ii) NLI models that measure entailment or contradiction in §4.2 and §4.3, and (iii) pretrained language models that form the model family $\mathcal{V}$ when computing PVI (in §4.2 and §4.4). To identify RCUs, we use out-of-the-box SRL models available in AllenNLP (Gardner et al., 2018; Shi and Lin, 2019) based on the BERT architecture (Devlin et al., 2019) (345M parameters). For detecting entailment or contradictions, we use a state-of-the-art NLI model (Laurer et al., 2022) with checkpoint available at Huggingface (Wolf et al., 2020).[6] We use the T5-large model (Raffel et al., 2020) as the model family $\mathcal{V}$ (770M parameters) finetuned on the gold reasoning chains (refer to paragraph below for details). Note that we use the original code for all text-generation metrics listed in §5.2. Specifically, rouge scores are computed using the python `rouge-score` package.

RCU Computation. As mentioned in §4.5, we use an SRL model to decompose a sentence into multiple 'subject-verb-object' frames. After obtaining a list of frames (often overlaping) from a sentence, we sort the frames by length and select a disjoint subset until any remaining frame is already contained in the sentence formed by the selected frames. From each frame, we remove modifiers (denoted by a separate tag) that contain a verb (checked using a PoS-tagging model from `nltk`) as it would also be identified as a separate frame. Once the RCUs are identified, we classify them into premise-RCUs or conclusion-RCUs based on the location in the sentence and rules based on the type of subordinating conjucntion (detected using PoS-tag). Typically, conclusion-RCU occurs at the very end of the sentence, but in case of 'because' or 'since' the RCU immediately following the conjunction is taken as the premise.

For instance, consider this example step from GSM-8K: "[The boots cost \$5 more than both pairs of heels together], so [the boots cost 99 + 5 = \$104]." Here, the two RCUs are joined using "so" and thus the first RCU is the premise and the second is the conclusion. In a different example, "[Allen's current age is 11/18*162 = 99] since [the fraction of the ratio that represents Allen's age is 11/18]." Here, the first RCU is the conclusion and the second one is the premise based on the conjunction "since". Even if the sentence began with "since", we would identified the RCU immediately following it to be the premise.

PVI Training. Similar to Chen et al. (2022), we use the T5-large model (Raffel et al., 2020) as the predictive model family $\mathcal{V}$ that is finetuned on gold reasoning chains using the train split of each dataset (with validation splits used for model selection). However, in our case, the model is trained to generate the conclusion-RCUs or the entire reasoning step (instead of the label in a classification task as done in Ethayarajh et al. (2022); Chen et al. (2022)). We compute log-probability over the text sequence as the length-normalized average of log-probabilities over all tokens (Brown et al., 2020). For intra-correct$_{\text{PVI}}$, g is a model trained to generate $y = \text{RCU}_c^{(i)}$ from $x = \text{RCU}_{\boldsymbol{p}}^{(i)}$ and g' is trained to generate $y = \text{RCU}_c^{(i)}$ directly. Using the train split of a reasoning dataset, we pool all steps from all reasoning chains. Each step is then decomposed into RCUs and constitutes one data point (x, y) for training the aforementioned models. The input to the model (used to generate y) could be template, i.e. "`[X] -> `", and "`None -> `", or a natural language sentence, "`[X], so `", and "`So, `" for g and g' respectively. Here, `[X]` represents the con-

[6] NLI model available at: https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

Input Context ($\mathcal{X}$)	**Gold Reasoning Chain**	**Orig. Perturbations**	**Our Perturbations**
The moon is a kind of moon. Earth is a kind of planet. Moons orbit planets. Gravity causes orbits. What keeps the Moon orbiting Earth?	Moon orbits planets and earth is a kind of planet, so moon orbits earth. Gravity causes orbits, so gravity causes the moon to orbit the earth.	Moon orbits planets and earth is not a planet, so moon orbits earth. Gravity causes orbits, so gravity causes the moon to orbit the earth.	Moon orbits planets and earth is a kind of planet, so moon does not orbit earth. Gravity causes orbits, so gravity causes the moon to orbit the earth.
Classifying means grouping objects by their properties. Shape is a property of appearance of an object. A galaxy is a kind of object. What feature is used to classify galaxies?	Classifying means grouping objects by their properties. Shape is a property of appearance of an object, so shape can be used to classify objects. A galaxy is a kind of object, so galaxies can be classified by shape.	Classifying means grouping objects by their properties. Comets orbits are elliptical, so shape can be used to classify objects. A galaxy is a kind of object, so galaxies can be classified by shape.	Classifying means grouping objects by their properties. Shape is a property of appearance of an object, so classification is a kind of process. A galaxy is a kind of object, so galaxies can be classified by shape.

Table 11: Differences in our perturbations to ones used in Golovneva et al. (2023) for errors NEG (top) and HALL (bottom). Overlapping text in input context and reasoning chains is underlined and perturbations are shown in red. For NEG with original perturbations, sentence embeddings of the perturbed overlapping sentence will be very different, leading to decrease in sentence similarity (does not occur in our perturbations). For HALL, shortcut is to check for facts missing from the input context by drop in sentence similarity (does not occur in our perturbations).

catenated premise units $\text{RCU}_{\boldsymbol{p}}^{(i)}$ (via 'and'). We find no significant change in performance when using the template or a natural language sentence. We use the latter to report performances in §6. For info-gain, the model g is trained to generate $y = \hat{a}$ given $[z, x] = s^{(\leq i)}$ and the training data are partial reasoning chains conditioned to generate the predicted answer. Since input to g' is $z = s^{(<i)}$, the input instances for g and g' overlap. Thus, we can use the same model for both g and g' as done by Chen et al. (2022). Note that $\hat{a}$ denotes the final answer sentence. So, $\hat{a}$ corresponds to the hypothesis sentence already provided in the EB dataset. In case of GSM-8K, we construct $\hat{a}$ by concatenating the question and the predicted answer, i.e., "`[Q] Answer: [A]`" where `[Q]`, and `[A]` are placeholders for question and predicted answer respectively. Throughout training the hyperparameters used are: learning-rate of $3e^{-5}$, 10 train epochs, with weight decay of 0.1 (all other hyperparameters are set to default). After training we select the model checkpoint (at epoch level) corresponding to the lowest 'rougeL' score on the validation split.

Range of Scores. Our $\text{intra-correct}_{\text{entail}}$ and inter-correct scores fall in the range $[0, 1]$ where 0 indicates failure and 1 indicates perfect score. By construction, PVI can be positive, negative or 0 which also applies to $\text{intra-correct}_{\text{PVI}}$ and $\text{info-gain}_{\text{PVI}}$. Positive PVI indicates a step is correct or informative, whereas negative (or zero) values indicate otherwise. Future works can explore normalization techniques to limit the range of these scores. Furthermore, informativeness of a step in a reasoning chain is an inherently subjective criteria which also depends on the underlying reasoning problem. Therefore, the $\text{info-gain}_{\text{PVI}}$ values of steps in different reasoning chains corresponding to different problem statements can be very different. Future work can also aim to address this variability.

B Datasets and Errors

We expand on the dataset descriptions provided in §5.1, and explain various error types. A glossary of error types is present in Table 16.

B.1 Entailment Bank

As described in §5.1, due to the construction of Entailment Bank, there is an overlap between $\mathcal{R}$ and $\mathcal{X}$. Therefore, if perturbations are applied to this overlapping information then it can spuriously lead to high correlation for any metric comparing $\mathcal{R}$ with $\mathcal{X}$ based on sentence-embeddings or n-grams. This happens because in gold or unperturbed chains there is high degree of overlap due to exact match and in the perturbed chains the overlap goes down significantly. However, if perturbations are applied to information not contained in $\mathcal{X}$, gold chains do not have high degree of overlap to begin with, and thus is a more challenging setting for evaluating metrics. Therefore, different from Golovneva et al. (2023), we only apply perturbations to facts/parts of the reasoning chain not in the input context.

We provide examples illustrating this phenomenon in Table 11. For negation errors, if we negate an overlapping source fact, comparing the chain with input the context leads to a direct drop in sentence similarity. We remove this shortcut by negating facts not contained in the input context. For hallucination errors, if a source fact is hallucinated, one can detect hallucinations by simply checking if a source fact is missing (drop in cumulative sentence similarity when compared to $\mathcal{X}$). We remove this shortcut by only applying hallucination perturbations to intermediate facts not in $\mathcal{X}$. Additionally, instead of sampling hallucinated text from other reasoning problems, we sample hallucinated text from irrelevant sentences or distractors provided for each instance in Entailment Bank (Task 2). This leads to higher word overlap between hallucinated text and input context.

Perturbations are first applied to intermediate nodes in the reasoning tree and then converted into a natural language reasoning chain. While borrowing error types from Golovneva et al. (2023), we make the following three additional changes: Firstly, the hallucinated text is sampled from distractors. Secondly, swap errors are introduced between the intermediate node and its parents, so that we can ensure incoherence in the reasoning chain. Thirdly, repetition errors are implemented by repeating an intermediate node twice (parent of the second node is the first node). Instead of verbatim repetition, we also introduce adding a paraphrase using a Pegasus-based model (Zhang et al., 2020)[7] and an irrelevant but true sentence to the reasoning chain. So in case of Figure 2(b), instead of verbatim repetition "the northern hemisphere is a kind of place", we would add text like "the norther hemisphere is a sort of location" and "daylight is when the sun shines" for PAR and RED errors respectively.

B.2 GSM-8K

We directly use the human-annotated reasoning chains for GSM-8K collected by Golovneva et al. (2023). We refer readers interested in the data collection process, and details about each error type to Appendix F of their paper (c.f. Table 15). In Table 13, we provide some examples of gold (human-written) reasoning chains in GSM-8K along with our identified RCU annotations. Note that while EB-challenge is constructed such that a perturbed reasoning chain only contains one error at a time, errors in GSM-8K dataset can co-occur as it contains model-generated errors that can be diverse.

[7]Checkpoint: https://huggingface.co/tuner007/pegasus_paraphrase

Method	Error Types			
	REP	HALL	NEG	SWAP
ROUGE-1	0.39	0.41	0.03	0.06
ROUGE-2	0.36	0.39	0.11	0.09
ROUGE-L	0.21	0.19	0.01	0.23
BERTScore	0.26	0.41	0.15	0.17
BARTScore	0.03	0.06	0.08	0.18
PRISM	0.23	0.45	0.03	0.16
CTC Relevancy	0.26	0.06	0.03	0.04
CTC Consistency	0.31	0.16	-0.05	-0.02
ROSCOE-SS (fine-tuned)	0.51	0.51	0.54	0.04
ROSCOE-SA (fine-tuned)	**0.82**	0.85	0.92	0.61
ROSCOE-LI	-0.04	0.40	0.91	-0.05
RECEVAL-correctness	0.09	**0.89**	**0.94**	**0.64**
RECEVAL-informativeness	0.79	0.31	0.04	0.10

Table 12: Comparison of Somer's D correlation scores using baseline text-generation metrics, ROSCOE, and our metrics on perturbations to Entailment Bank by Golovneva et al. (2023).

C Evaluating RECEVAL on EB-regular

We evaluate the performance of all metrics on the originally perturbed sentences (EB-regular) in Table 12. While the relative trends between RECEVAL and other baselines remain the same, we find that ROSCOE's correlation values on HALL, NEG and SWAP are much higher than Table 1a where the aforementioned shortcuts do not exist. Furthermore, correlation values of text-generation metrics on HALL errors also decrease when spurious shortcuts are removed. Nevertheless, RECEVAL outperforms baselines on correctness errors. Note that we do not consider grammar, missing errors from Golovneva et al. (2023). This is mainly because missing steps involve a confounder and are hard to evaluate in a reference-free manner. Further, grammar issues that do not alter correctness can be measured easily by grammar-checking metrics used in ROSCOE-LC.

D RECEVAL Correctness Metrics

In this section, we provide additional details and ablations about the correctness metrics in RECEVAL as discussed in §6.2.

Oracle RCUs. In §6.2, we evaluate our identified RCUs with gold RCUs using entailment trees from Entailment Bank. Given an intermediate node, we decompose it into RCUs by picking the largest

Input Context	Reasoning Chain
Tina buys 3 12-packs of soda for a party. Including Tina, 6 people are at the party. Half of the people at the party have 3 sodas each, 2 of the people have 4, and 1 person has 5. How many sodas are left over when the party is over?	[Tina buys 3 12-packs of soda for 3*12=36 sodas]. [6 people attend the party], so [half of them is 6/2= 3 people]. [Each of those people drinks 3 sodas], so [they drink 3*3=9 sodas]. [Two people drink 4 sodas], which means [they drink 2*4=8 sodas]. With [1 person drinking 5], [that brings the total drank to 5+9+8+3=25 sodas]. As [Tina started off with 36 sodas], that means [there are 36-25=11 sodas left].
Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens?	If [each chicken eats 3 cups of feed per day], then for 20 chickens [they would need 3*20=60 cups of feed per day]. If [she feeds the flock 15 cups of feed in the morning], and [25 cups in the afternoon], then [the final meal would require 60-15-25=20 cups of chicken feed].

Table 13: Example of reasoning chains in GSM-8K dataset with identified RCUs shown in parenthesis ('[.]').

Method	Error Types		
	HALL	NEG	SWAP
inter-correct	0.14	0.90	0.16
inter-correct (+ premises)	0.15	0.87	0.13
inter-correct$_{\text{concat}}$	0.14	0.89	0.22

Table 14: Comparison of different variants of inter-correct metric by including premises and concatenation instead of pair-wise comparison on validation split of EB-challenge.

Method	Previous	Error Types		
	Steps (k)	HALL	NEG	SWAP
inter-correct$_{\text{no-contr.}}$	all	0.14	**0.89**	0.22
inter-correct$_{\text{no-contr.}}$	2	0.10	<u>0.84</u>	0.20
inter-correct$_{\text{entail}}$	2	0.56	0.73	0.32
inter-correct$_{\text{PVI}}$	2	0.84	0.10	0.34
inter-correct$_{\text{no-contr.}}$	1	0.08	0.79	0.15
inter-correct$_{\text{entail}}$	1	0.52	0.66	0.31
inter-correct$_{\text{PVI}}$	1	0.81	0.05	0.26
intra-correct$_{\text{no-contr.}}$	0	0.02	0.82	0.08
intra-correct$_{\text{entail}}$	0	0.71	<u>0.84</u>	<u>0.37</u>
intra-correct$_{\text{PVI}}$	0	**0.86**	0.16	**0.38**

Table 15: Comparison of different views of correctness based on current step and preceding k steps on validation split of EB-challenge. Note that inter-correct$_{\text{no-contr.}}$ is same as inter-correct$_{\text{concat}}$.

SRL frame (including modifiers). For the premise-RCUs, we find all RCUs from its parent nodes. This ensures that all the premise-RCUs used to form the conclusion are included when measuring correctness and avoids any irrelevant sentences (which are neutral when measuring entailment and independent from an information-theoretic perspective). This explains why using gold RCUs boosts the performance on intra-step-correctness.

Variants of inter-correct. As described in §4.3, we perform pair-wise comparison wit all prior information in $\mathcal{X}$ and conclusion-RCUs from preceding steps. Due to high overlap in information contained in premise-RCUs and $\mathcal{X}$, we did not measure correctness with respect to premises. Alternative to pair-wise comparison, one can also concatenate all prior information and check for contradiction directly (denoted by inter-correct$_{\text{concat}}$). We compare these three different implementations of inter-step correctness in Table 14. We find that the performance of concatenation and pair-wise variants is comparable across all error types. As expected, we observe similar performance of inter-step correctness when including premise-RCUs across all errors.

Different views of correctness. In §4.2 and §4.3, we present three views of correctness: (i) entailment, (ii) using PVI framework, and (iii) lack of contradictions. The first two are used to compute intra-correct and the last is used to compute inter-correct. Then in §6.2, we compare all three views of correctness to compute intra-correct and conclude intra-correct$_{\text{PVI}}$, and intra-correct$_{\text{entail}}$ work best with hallucination and negate errors respectively (with comparable performance on swap). Now, we extend this analysis to evaluate how these three views of correctness compare when evaluating inter-step correctness in Table 15. Since PVI and entailment variants concatenate information, to maintain uniformity, we use inter-correct$_{\text{concat}}$ for this analysis. First, we observe that the best performance on nega-

Error	Dataset	Description	Correctness	Informativeness
HALL	EB, GSM-8K	Hallucinations: Step contains information not provided in the input context, could be irrelevant but makes the step wrong.	✓	✗
REP	EB, GSM-8K	For EB: Step contains verbatim repetition of information already in previous steps. For GSM-8K: Step contains verbatim repetition or paraphrasing of information already present. The step could be dropped without impacting correctness.	✗	✓
RED	EB, GSM-8K	Additional step in the reasoning chain containing information irrelevant to solving the problem. The information itself could be factual and consistent with input context.	✗	✓
PAR	EB	Additional step contains paraphrasing of information already in the reasoning chain.	✗	✓
NEG	EB	Compared to the gold chain, step contains negation of information altering the correctness.	✓	✗
SWAP	EB	Information within the step is swapped in order, altering the overall correctness.	✓	✗
QUAL	GSM-8K	Likert score (1-5), measures overall quality of reasoning chain and how well it answers the question.	✓	✓
COH	GSM-8K	Likert score (1-5), measures overall coherence of the reasoning chain, i.e. if it makes sense and is non-contradictory.	✓	✓
COM	GSM-8K	If the step contains any commonsense or general world knowledge related mistake.	✓	✗
FACT	GSM-8K	Step contains information that contradicts some information in the input context.	✓	✗
LOGIC	GSM-8K	Step contains errors in logical deduction, could be contradictory to previous steps or not enough support or evidence, relates to coherence.	✓	✗
MATH	GSM-8K	Arithmetic or math equation errors in the step.	✓	✗

Table 16: Glossary of types of errors in EB-challenge and GSM-8K and how it relates to desired correctness and informativeness properties of good reasoning chains. Note that '✓' and '✗' denote the expected impact on correctness and informativeness in general. The actual impact depends on the reasoning chain and the exact error.

tion errors is obtained by $\text{inter-correct}_{\text{no-contr.}}$ with $k = all$, whereas for the rest best performance is obtained using $\text{intra-correct}_{\text{PVI}}$ ($k = 0$). Further, we find that $\text{inter-correct}_{\text{PVI}}$ works best to identify hallucinations (and swaps), whereas $\text{inter-correct}_{\text{no-contr.}}$ is best for negation across all values of k. Lastly, $\text{inter-correct}_{\text{entail}}$ correlates well across error types for different values of k. This leads to a unified correctness metric wherein different methods differ in the view of correctness employed and the number of preceding steps k considered.

E Approximately Positive Information Gain (API)

In §6.3, we introduce API to quantify the trend of informativeness across steps in a reasoning chain. A reasoning chain is API_k across steps if for every k contiguous steps, these steps as a whole are more informative than the preceding steps. Based on the PVI framework, a reasoning chain would be API_k if $\text{PVI}(s^{(i:i+k-1)} \rightarrow \hat{a}|s^{(<i)}) > 0, \forall s^{(i)} \in \mathcal{R}$. Below we show how to evaluate this quantity directly in terms of our metric $\text{info-gain}_{\text{PVI}}$.

$$
\begin{aligned}
&\text{PVI}(s^{(i:i+k-1)} \rightarrow \hat{a}|s^{(<i)}) \\
&= \log g[s^{(<i+k)}](\hat{a}) - \log g[s^{(<i)}](\hat{a}) \; (\because g = g') \\
&= \log g[s^{(<i+k)}](\hat{a}) - \log g[s^{(<i+k-1)}](\hat{a}) \\
&\quad + \log g[s^{(<i+k-1)}](\hat{a}) \cdots - \log g[s^{(<i+1)}](\hat{a}) \\
&\quad + \log g[s^{(<i+1)}](\hat{a}) - \log g[s^{(<i)}](\hat{a}) \\
&= \text{PVI}(s^{(i+k-1)} \rightarrow \hat{a}|s^{(<i+k-2)}) + \cdots \\
&\quad + \text{PVI}(s^{(i)} \rightarrow \hat{a}|s^{(<i)}) \; \text{(using definition in §4.4)} \\
&= \sum_{j=i}^{i+k-1} \text{info-gain}_{\text{PVI}}^{(j)}
\end{aligned}
$$