COKE: Customizable Fine-Grained Story Evaluation via Chain-of-Keyword Rationalization

Brihi Joshi[1*] **Sriram Venkatapathy**[2] **Mohit Bansal**[2] **Nanyun Peng**[2] **Haw-Shiuan Chang**[3*]
[1]University of Southern California, [2]Amazon AGI Foundations,
[3]University of Massachusets Amherst
brihijos@usc.edu
{vesriram,mobansal,pengnany}@amazon.com
hschang@cics.umass.edu

Abstract

Evaluating creative text such as human-written stories using language models has always been a challenging task – owing to the subjectivity of multi-annotator ratings. To mimic the thinking process of humans, chain of thought (Wei et al., 2023) (CoT) generates free-text explanations that help guide a model's predictions and Self-Consistency (Wang et al., 2022) (SC) marginalizes predictions over multiple generated explanations. In this study, we discover that the widely-used self-consistency reasoning methods cause suboptimal results due to an objective mismatch between generating 'fluent-looking' explanations vs. actually leading to a good rating prediction for an aspect of a story. To overcome this challenge, we propose **C**hain-**o**f-**Ke**ywords (COKE), that generates a sequence of keywords *before* generating a free-text rationale, that guide the rating prediction of our evaluation language model. Then, we generate a diverse set of such keywords, and aggregate the scores corresponding to these generations. On the StoryER dataset, COKE based on our small fine-tuned evaluation models not only reach human-level performance and significantly outperform GPT-4 with a 2x boost in correlation with human annotators, but also requires drastically less # of parameters.

1 Introduction

Evaluating stories is an important and time-consuming job for professionals in the entertainment industry. For example, novel competition judges, book editors, or movie producers might need to select the best story from thousands of submissions according to their tastes and the understanding of the market.

As LLMs get better at judging story quality, automatically evaluating human-written stories becomes practical. However, there are still several challenges to overcome. First, judgements from off-the-shelf LLMs might be biased towards the preference of particular annotators during the alignment stage, which could be very different from the tastes of the desired population. Second, humans are extremely subjective in judging creative writing like stories, which is often demonstrated in their creativity: Some readers or professional reviewers would think character shaping is the most critical component for evaluating a story, whereas others might like or dislike the characters along with some other components, like the scene description mentioned in the story. This lack of consensus in likes and dislikes, along with differences across aspects (e.g. character shaping, ending, etc) in the story makes evaluating human-written stories an extremely difficult task.

The desired human evaluation here would entail that we collect diverse opinions from different readers/reviewers to estimate a *average* opinion of the story from a desired population, but this is extremely tedious and expensive. This high cost has motivated automatic measures for evaluating the stories written by humans. In this study, we aim at building an automatic story evaluation system that can 1) provide fine-grained evaluation for a human-written story in predefined and/or customized aspects, 2) provide a set of *rationales* that model diverse opinions of multiple humans and help us better predict the average score for different aspects of the story, and 3) be easily customized toward the opinions of the desired population (i.e., fine-tunable using the collected human judgements and explanation).

The reason-then-predict approaches like Chain of Thought (CoT) (Wei et al., 2023) not only improve the interpretability of the said predictions by generating rationales but also improve downstream performance in predictions (Wei et al., 2023; Wang et al., 2023b). Using these approaches, Large Language Models (LLMs) can score arbitrary aspects of a story without any additional training. How-

[*]Work is mostly done at Amazon

Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM² 2025), pages 366–384
July 31 - August 1, 2025

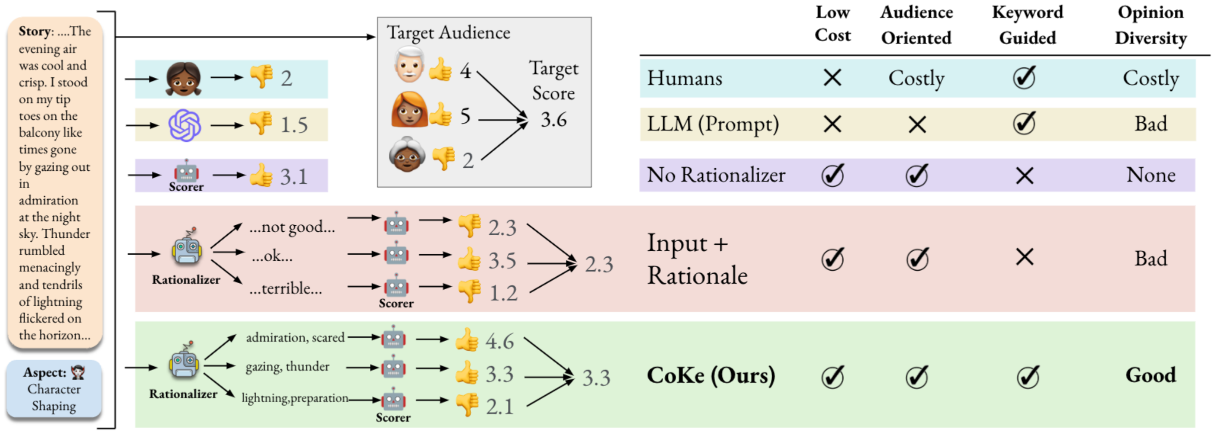

Figure 1: CoKE provides a low-cost, audience-oriented (customizable), and keyword-guided approach to evaluating stories by generating and scoring diverse keyword sequences that explain a fine-grained aspect-story pair.

ever, for story evaluation particularly, the scores from prompting LLMs might deviate from the population average of our target audience, along with significant cost induced by large token lengths of such inputs.

Fine-tuning a small Language Model (LM) to directly predict the population average of annotators is a cheap viable alternative, but does not provide rationales while also being inflexible w.r.t. how granular we want the story to be evaluated (e.g., character shaping of the vampire, ending w.r.t. a certain character, etc). Another option is fine-tuning a small LM to generate free-text rationals for CoTs and use the self-consistency (Wang et al., 2022) approaches to marginalize over multiple sampled CoTs. However, we discover that the free-text rationals tend to reduce the diversity of CoTs' rating predictions and deviate the average prediction rating from the population average.

In order to mitigate this shortcoming we propose Chain-of-Keywords, CoKE, which consists of two simple yet effective modifications to regular CoT approaches. First, instead of just generating a free-text rationale, we generate a chain of *keywords* before generating a rationale that can describe salient concepts in and outside the story. Our intuition is that keywords help prevent the learning and generation of annotator artifacts (like sentiment-laden words and other personal descriptors like 'I think, I feel', etc), which assists with the objective misalignment we see in CoT approaches. Like SC, instead of generating one rationale, it samples *multiple* keyword rationales, which simulates annotator diversity and helps better estimate the population average. Therefore, CoKE uses the generated keywords to score a story, and the corresponding generated rationale for interpreting the story, as shown in Figure 1.

On StoryER (Chen et al., 2022), a fine-grained story evaluation benchmark (Chen et al., 2022), we show that CoKE can better estimate population averages as compared to LLM baselines using GPT-3.5 (text-davinci-003) and GPT-4 (gpt-4-0613) (Brown et al., 2020; Ouyang et al., 2022), as well as open-source LLMs like LLaMa-2-7B-Chat (Touvron et al., 2023) and Mistral-7B-Instruct (Jiang et al., 2023). We also show that CoKE consistently outperforms self-consistency and approaches based on supervised fine-tuning, including those where the rationale generated is specifically aligned to that of annotator-written explanations using reinforcement learning (RL), as well as improved correlations on human evaluations as compared to baselines. Furthermore, we also show that CoKE can work effectively even when built on smaller LMs as its backbone (approx. 58x fewer # of parameters than GPT-3.5), while surpassing GPT-3.5 by 2.18x improvement in correlation metrics with the target annotator population. To the best of our knowledge, CoKE is a first rationalize-then-predict approach for fine-grained story evaluation surpassing LLM performance for this task, and reaches human-level performance in the StoryER dataset (Chen et al., 2022).[1]

2 Problem Formulation

We begin by describing our task setup and why the task is challenging.

[1]Our code and models will be released.

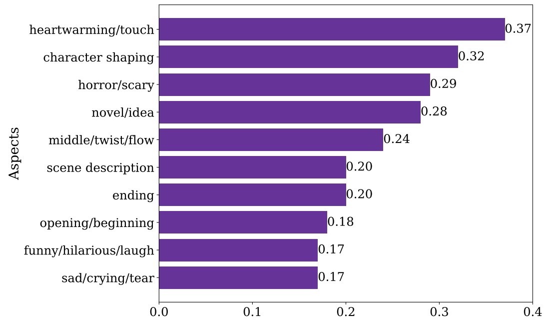

Figure 2: ICC annotator agreements scores for the stories with a certain aspect in the training set.

Task setup. We are given a story, along with an aspect, with respect to which we want to evaluate the story. The aspect can focus on certain semantic or literary features of the story (Gülich and Quasthoff, 1986), like *humor*, *character shaping*, etc. Our task is to evaluate the story with respect to the given aspect and provide a Likert rating between 1 and 5, where a higher score implies the story is better with respect to the aspect.

We assume that there exists a dataset that consists of the story-aspect ratings and the explanations for the ratings. One story-aspect pair could be annotated by multiple annotators from our target audience. Any automated story evaluation system should provide a single score for an aspect-story pair that is close to the average ratings from annotators, without modeling the individual annotator (Sap et al., 2021; Wang et al., 2023a).

Story evaluation is an extremely subjective task. We use the StoryER dataset (Chen et al., 2022) for our task. What is interesting to note here is even though all annotators have to focus on a certain aspect of the story, human ratings are still extremely subjective. In the StoryER dataset, we calculate Intraclass Correlation Coefficient (ICC) scores (Cicchetti, 1994) to evaluate annotator agreements within annotators for a given aspect, across all the possible stories which are marked with that aspect (Figure 2). The 'heartwarming' aspect has the highest agreement of 0.37, which is still considered to be poor while interpreting ICC scores (Cicchetti, 1994).

Limitation of CoTs for story evaluation. Self-consistency (Wang et al., 2022) is an approach that extends Chain of Thought (CoT) (Wei et al., 2023) to capture the diverse opinions of humans. Wang et al. (2022) sample various free-text rationales and marginalize the different predictions based on the generated CoT. However, it is very difficult to decode all possible rationales. Furthermore, there could be some objective misalignments between generating highly probable and *coherent* rationales and predicting the final ratings from annotators (Jia et al., 2020). For example, let's say in our training data, our vampire stories and their corresponding explanations are all good and positive. Then, if there are some vampire stories that are boring and contain some grammatical errors during the testing test time, the LM does not know how to generate a negative rationale for a vampire story, so it is forced to generate coherent but biased rationales, which lead to positive rating predictions.

3 Chain-of-Keywords (CoKe)

There are three kinds of words in a free-text explanation: sentiment words, keywords referring to the concepts in the story, and the functional words (e.g., stop words). We view the sentiment and functional words as an artifact for story evaluation because they only provide the information that the rating has already provided and could induce a bias in CoT's rating prediction. This is because the probability of generating a positive sentiment word might be affected more by the nearby function words than by the quality of the input story and thus, the positive sentiment in the explanation would heavily bias the CoT to predict a high score.

For example, we observe that most positive rationales in the StoryER dataset are much more likely to contain "*I*" while the most negative rationales have much more "*It*". In the positive rationales, *I* is the 8th likely words (1.8%) while *It* is the 14th likely words (0.6%). In the negative rationales, *I* is the 16th likely words (0.9%) while *It* is the 7th likely words (1.5%). If we observe some rationales starting with "*I like*" or "*I love*" in the training vampire stories, "*I*" could become the most likely first word in the generated rationale for a bad testing vampire story, which bias the CoT to output *like/love* and a high rating at the end.

We leverage these intuitions to build CoKe in the following manner (shown in Figure 3). First, a language model is fine-tuned to generate *keywords*, along with a free-text explanation conditioned on those keywords, that inspects the story w.r.t the aspect. These keywords are in the form of phrases (from the story itself) that specifically do not contain artifacts. From this language model's decoder,

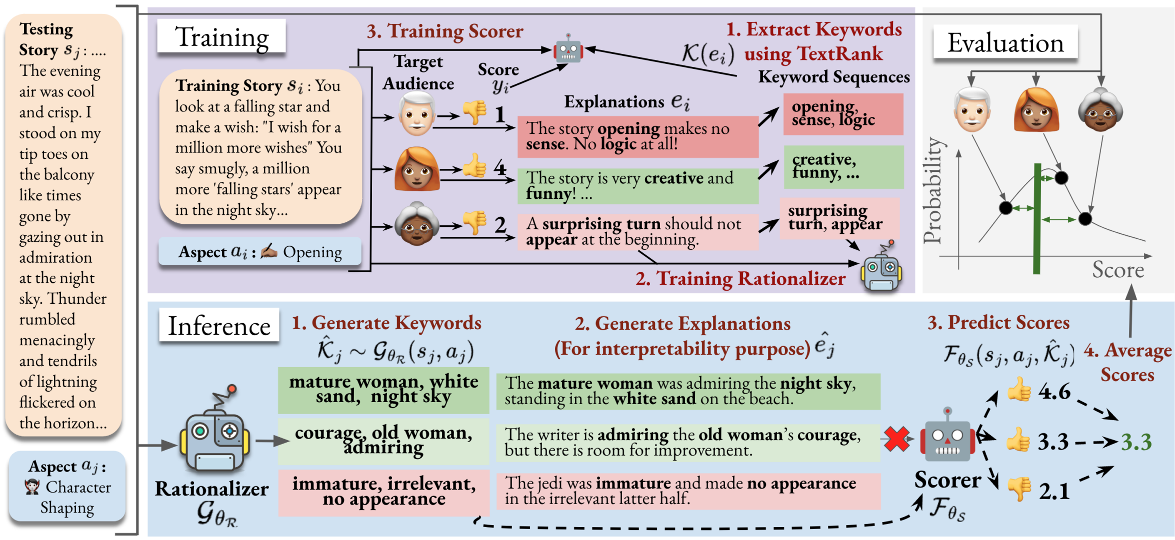

Figure 3: During training, CoKE extracts keywords from annotator explanations and train rationalizers and scorers. During inference, CoKE first samples candidate keyword sequences (for the scorer) and explanations (for better interpretability), and then score the individual generated candidates before aggregating them. Our purpose is to obtain a better *population average* that can capture diverse annotator scores.

we sample multiple keyword sequences, intended to simulate diverse annotator opinions. A trained scorer model is then used to produce score predictions from aspect-story-keyword triples, and scores for all individual candidate keyword sequences are averaged to produce the final score.

More concretely, let $\mathcal{D}_{Tr}$ and $\mathcal{D}_{Te}$ be the training and test datasets respectively. They are composed of the story-aspect-explanation-rating tuple (s_i, a_i, e_i, y_i). For example, in StoryER, s_i is a human-written story from WritingPrompts (Fan et al., 2018), a_i is one of the predefined aspects, y_i is the rating from an annotator, and e_i is the text justification for y_i. If two annotators label the same story and aspect, the s_i and a_i would be the same for the two tuples.

CoKE consists of two components: a rationalizer model, θ_R, and a scorer model, θ_S. The rationalizer is a seq2seq language model that is fine-tuned to generate rationales, given an aspect-story pair as an input: $\hat{\mathcal{K}}_j \sim \mathcal{G}_{\theta_\mathcal{R}}(s_j, a_j)$, while the scorer is a regression language model that is fine-tuned to predict a floating point score, given aspect-story-rationale triplets as an input: $y_j = \mathcal{F}_{\theta_S}(s_j, a_j, \hat{\mathcal{K}}_j)$. We detail the training and inference process of CoKE below and further conduct ablations on different components of CoKE to justify our keyword extraction step and other design decisions in Section 4.5.

Training in CoKE. Given story-aspect-explanation-rating tuple (s_i, a_i, e_i, y_i), we first extract the *keywords* from the annotator-written explanation e_i and train our rationalizer to first generate the extracted keyword sequence $\mathcal{K}(e_i)$ before generating the explanation e_i. We template the inputs for the rationalizer to contain both the aspect and story - `aspect: <aspect> story: <story>`, and the output is a chain of keywords, followed by a free-text explanation that is conditioned on the keywords, which looks like - `keywords: <key1, key2, ..., keyn> rationale: <natural language explanation>`.

For the scorer, we provide the story s_i, aspect a_i, and extracted keyword sequence $\mathcal{K}(e_i)$ as the input and ask it to predict the rating from the annotator y_i. The input to the model looks like - `aspect: <aspect> story: <story> keywords: <keywords>` and the loss function is the mean squared error.

Inference in CoKE. After training θ_R and θ_S separately, CoKE inference is explained below.

We simulate diversity in annotators by sampling multiple candidate keyword sequences using $\mathcal{G}_{\theta_\mathcal{R}}$, and then marginalize the candidate rationales by taking a mean over scores of individual candidates. This score is represented as follows -

$$\mathbb{E}_{\hat{\mathcal{K}}_j \sim \mathcal{G}_{\theta_\mathcal{R}}(s_j, a_j)} \left[\mathcal{F}_{\theta_S}(s_j, a_j, \hat{\mathcal{K}}_j) \right], \quad (1)$$

where (s_j, a_j) is a testing example from $\mathcal{D}_{Te}$.

Since finding all possible $\hat{\mathcal{K}}_j$ is not feasible to calculate the expectation term, we conduct Monte Carlo simulations over a set number of samples,

Setting	Rationale for Scorer	Rationalizer	Scorer	Metrics		
				Pearson's ρ (↑)	MSE (↓)	F1-Score (↑)
LLM	-	None	GPT-3.5	0.0240	0.5172	0.2277
	-	None	GPT-3.5 5-shot	0.1440	0.2703	0.4751
	Explanation	GPT-3.5 CoT		0.1049	0.2290	0.4833
	Explanation	GPT-3.5 CoT SC Mean		0.1303	0.1970	0.5267
	Explanation	GPT-4 CoT		0.1093	0.3039	0.4199
	Explanation	Mistral-7B-Instruct CoT		0.0573	0.5113	0.3760
	Explanation	Mistral-7B-Instruct CoT 5-shot		0.0596	0.5019	0.3760
	Explanation	Mistral-7B-Instruct CoT-SC MV		0.0648	0.5252	0.3760
	Explanation	Mistral-7B-Instruct CoT-SC Mean		0.1023	0.4998	0.3740
	Explanation	Mistral-7B-Instruct CoT-SC Mean 5-shot		0.1266	0.4578	0.3940
	Keywords	Mistral-7B-Instruct CoT		0.0277	0.6892	0.2007
	Keywords	Mistral-7B-Instruct CoT 5-shot		0.0300	0.6676	0.2101
Supervised Fine-tuning	Explanation	T5-Small	DeBERTa-V3-Small	0.0904	0.1339	0.5827
	Explanation	T5-Small PPO	DeBERTa-V3-Small	0.0779	0.1118	0.5773
	Explanation	T5-Small CoT		0.0676	0.1698	0.5622
	-	None	T5-Small	0.0712	0.1620	0.5647
	-	None	T5-Small Prob-avg	0.2451	0.1331	0.6162
Human	Explanation	Human		0.3037	0.1972	0.4998
CoKe	Keywords	T5-Small	DeBERTa-V3-Small	**0.2900**	**0.0912**	**0.6334**
	Keywords	T5-3B	DeBERTa-V3-Small	**0.3142**	**0.0811**	**0.6509**

Table 1: We compare CoKe to other baselines that use rationalize-then-predict paradigms in StoryER. For all Self-Consistency (SC) variations, we average over 40 samples as done by (Wang et al., 2022). For CoKe, we provide the best performing setting with $\mathcal{N}$ = 100 samples.

$\mathcal{N}$, over which we average the score. Notice that $\mathcal{G}_{\theta_{\mathcal{R}}}$ could also generate the free-text explanations, $\hat{e}_j$, after the keywords, but they are just for interpretability purpose and won't affect the final score prediction.

4 Experiments

In this section, we evaluate CoKe, LLMs with sophisticated inference strategies, supervised fine-tuning, along with CoKe ablations.

4.1 Evaluation Setup

We train our T5 (Raffel et al., 2023) rationalizer and DeBERTa-V3 (He et al., 2021) scorer using the training set of StoryER (Chen et al., 2022) dataset and evaluate CoKe using its official test set. We first filter out story-aspects pairs that are only rated by one annotator and normalize the scores from annotators and models into the range from 0 to 1, using min-max normalization where max=5 and min=1. Given an input story-aspect pair, each model can only produce a single score. As shown in the evaluation block of Figure 3, we compare the output score with each annotator-provided score separately and the prediction that is closer to the average of all the human scores would perform better. This procedure allows us to compare each model with human performance and handle the varying numbers of human annotators, given the same input pair in StoryER.

We report three metrics for every evaluation conducted – Pearson's Correlation Coefficient (ρ), Mean Squared Error (MSE), and F1-score on binarized score values, thresholded using a value of 0.5. We use the Pearson correlation coefficient as the main metric because the global score average might be very different for different human annotators or different models. For example, the GPT-4's scores are found to be over-generous sometimes (Doostmohammadi et al., 2024; Gmyrek et al., 2024).

4.2 Human vs. CoKe

To estimate human performance, we use one annotator as the *prediction* that is compared to the other annotators for each pair of story and aspect. This process is repeated for every annotator's rating and story-aspect pair. In Table 1, we see that CoKe's best configuration significantly outperforms the human performance in MSE and slightly in Pearson's ρ, which shows that CoKe's prediction is closer to the population average than the individual human.

4.3 LLMs vs. CoKe

We prompt a mix of closed and open-sourced Large Language Models like **GPT-3.5** (text-davinci-003) and **GPT-4** (gpt-4-0613) (Brown et al., 2020; Ouyang et al., 2022; OpenAI et al., 2024), and **Mistral 7B Instruct** (Jiang et al., 2023) to generate a score for a given story-aspect pair. These models can be prompted to generate a score as-is or with a rationale, with the help of Chain of Thought (CoT) prompting (Wei et al., 2023). We evaluate zero- and few-shot prompting *without* CoT and *with* CoT. As seen in Table 1, our approach always outperforms strong LLMs prompted with CoT prompts to score an aspect-story pair. We can note a 3x improvement in Pearson's ρ shown by CoKE ($\approx$3B) in comparison to GPT-3.5 CoT while having an estimated 58x lesser number of parameters that GPT-3.5 ($\approx$175B).

We also run Self-Consistency (SC) approaches as shown by (Wang et al., 2022). We generate 40 CoT predictions per story-aspect pair in the test set and show two variations to aggregate scores provided by these CoTs: **Majority Voting** (MV) and **Mean**, a more suitable method for story evaluation tasks. Table 1 shows that CoKE correlates with the population averages better than the SC approaches. Appendix A further demonstrates that CoKE also outputs much more diverse ratings than SC.

4.4 Supervised Fine-tuning (SFT) vs. CoKE

Rationalization approaches pre-dating LLMs also fine-tuned smaller LMs to generate rationales, and then predict an answer based on the rationale and the input (Wiegreffe et al., 2021; Marasović et al., 2022). The approaches are cost-efficient and could be easily customized for the target audience. We use the *pipeline approach* (Wiegreffe et al., 2021) for generating both the rationales and scores for a given aspect-story pair (**T5-small + DeBERTa-V3-Small**). The *pipeline* is the same as CoKE except that T5 generates only one free-text explanation rather than multiple keyword sequences (i.e., $\mathcal{N}$ = 1 and $\mathcal{K}(\cdot) = \mathbb{1}(\cdot)$).

A shortcoming of the *pipeline* approaches is that they do not focus on the quality of the rationales that are generated. To mitigate the explanation distribution mismatch (Kirk et al., 2024) between annotators and generation, we added an additional alignment step, where generated rationales would be compared to the annotator-provided explanations using a Cider score reward (Vedantam et al., 2015), and used as feedback into the RATIONALIZER using the PPO algorithm (Schulman et al., 2017; Ramamurthy et al., 2022) (**T5-small PPO + DeBERTa-V3-Small**). Surprisingly, in Table 1 we see that specifically aligning generations with annotated explanations does not aid downstream scoring performance. This validates that explicitly improving rationale quality does not improve downstream

Rationalizer	Scorer	Metrics
		Pearson
-	$(s, a) \rightarrow$ DeBERTa-V3 Small	0.2718
-	$(s, a) \rightarrow$ DeBERTa-V3 Large	0.2697
T5 Small $\rightarrow (e)$	$(s, a, e) \rightarrow$ DeBERTa-V3 Small	0.2040
T5 Small $\rightarrow (e)$	$(a, e) \rightarrow$ DeBERTa-V3 Small	0.1912
T5 Small $\rightarrow (\mathcal{K}_{\text{TF-IDF}}(e))$	$(s, a, \mathcal{K}_{\text{TF-IDF}}(e)) \rightarrow$ DeBERTa-V3 Small	0.2548
T5 Small $\rightarrow (\mathcal{K}_{\text{Rake}}(e))$	$(s, a, \mathcal{K}_{\text{Rake}}(e)) \rightarrow$ DeBERTa-V3 Small	0.2081
T5 Small $\rightarrow (\mathcal{K}_{\text{TextRank}}(e))$	$(s, a, \mathcal{K}_{\text{TextRank}}(e)) \rightarrow$ DeBERTa-V3 Small	0.2727
T5 Small $\rightarrow (\mathcal{K}_{\text{TextRank}}(e))$	$(a, \mathcal{K}_{\text{TextRank}}(e)) \rightarrow$ DeBERTa-V3 Small	0.1924
T5 Small $\rightarrow (\mathcal{K}_{\text{TextRank}}(e), e)$	$(s, a, \mathcal{K}_{\text{TextRank}}(e)) \rightarrow$ DeBERTa-V3 Small	0.2800
T5 Large $\rightarrow (\mathcal{K}_{\text{TextRank}}(e), e)$	$(s, a, \mathcal{K}_{\text{TextRank}}(e)) \rightarrow$ DeBERTa-V3 Small	0.2834
T5 3B $\rightarrow (\mathcal{K}_{\text{TextRank}}(e), e)$	$(s, a, \mathcal{K}_{\text{TextRank}}(e)) \rightarrow$ DeBERTa-V3 Small	0.2887
T5 3B $\rightarrow (\mathcal{K}_{\text{TextRank}}(e), e), \mathcal{N} = 100$	$(s, a, \mathcal{K}_{\text{TextRank}}(e)) \rightarrow$ DeBERTa-V3 Small	**0.3142**

Table 2: Ablation study. s is a story, a is an aspect, e is an explanation, and $\mathcal{K}(.)$ is a keyword extraction function. For rationalizers, $\mathcal{N}$ = 10 except for the last row. CoKE (Ours) in the last four rows are highlighted.

aspect-story evaluation (Kirk et al., 2024; Florian et al., 2024).

In another approach, we fine-tune a T5 model to first generate an explanation, followed by a score (**T5-small CoT**) (Kim et al., 2023) without training another scorer model. Table 1 shows that SFT approaches are not at par with LLM-based baselines, and thus by default, lag behind CoKE. Based on Marasović et al. (2022), we also make a modification to SFT-CoT, where instead of generating a score conditioned on the explanation, we generate the score before generating the explanation (**T5-small**) (Marasović et al., 2022). Instead of sampling score, we also calculate *expected* predicted score for which we compute the weighted average according to the probabilities of each score token (**T5-small Prob-avg**). This leads to significant improvements in Pearson's ρ over other SFT approaches in Table 1, which shows the importance of generation diversity in this task.

4.5 CoKE Ablations

No Rationalizer in CoKE. During inference, CoKE's scorer takes in the aspect-story pair, along with the generated keywords from a fine-tuned rationalizer model. Here, we remove the rationales from the input of the scorer and fine-tune DeBERTa-V3 models to predict a score only based on the aspect-story pair (s,a). In Table 2, we see that the $(\mathbf{s}, \mathbf{a}) \rightarrow$ **DeBERTa-V3 Small/Large** baselines are strong, surpassing performances by LLMs in Table 1, while being significantly worse than CoKE. Furthermore, it cannot provide rationales or consider the user-specified aspects/keywords.

Varying Rationales in CoKE. In Section 3, we use $\mathcal{K}(\cdot)$ to extract keywords from the gold explanations e in the dataset (during training of

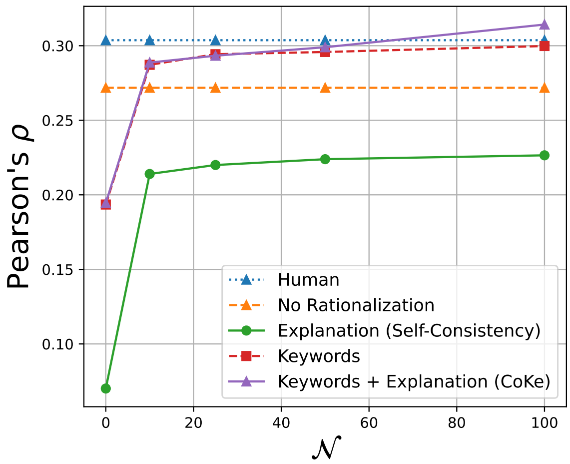

Figure 4: Pearson's ρ increases with the larger number of candidate generations ($\mathcal{N}$) in COKE and it's ablations. The rationalizer model here is T5-3b. We note that increasing the diversity of generation helps with better estimation of population preferences.

the rationalizer and scorer). First, we remove the keyword extraction step $\mathcal{K}(\cdot)$ in the baseline $(\mathbf{s}, \mathbf{a}, \mathbf{e}) \rightarrow$ **DeBERTa-V3 Small** to verify the design. This is equivalent to **T5-small + DeBERTa-V3-Small** in Table 1, except that we use $\mathcal{N} = 10$ rather than $\mathcal{N} = 100$ here. Its ρ (0.2040) is much worse than the ρ of $(\mathbf{s}, \mathbf{a}) \rightarrow$ **DeBERTa-V3 Small** (0.2718). To investigate the reason, we conduct another baseline that removes the story, the most important signal, from the input of the scorer ($(\mathbf{a}, \mathbf{e}) \rightarrow$ **DeBERTa-V3 Small**) and we find that its ρ only degrades slightly to 0.1912. This indicates that the scorer relies too much on the signal in explanation (e.g., sentiment words) to predict the ratings and ignore the signal in the story itself.

We also try different keyword extractors: TF-IDF (Frank et al., 1999), Rake (Rose et al., 2010) and TextRank (Mihalcea and Tarau, 2004). After keyword extraction, we remove all sentiment words from the keyword sequence. In COKE, we use TextRank for our choice of $\mathcal{K}(\cdot)$ due to its best performance in Table 2.

Finally, we find **T5 Small** $\rightarrow (\mathcal{K}_{\text{TextRank}}(\mathbf{e}), \mathbf{e})$ in COKE (0.2800) slightly outperforms **T5 Small** $\rightarrow (\mathcal{K}_{\text{TextRank}}(\mathbf{e}))$ (0.2727), which implies that predict the free-text explanations after keywords further improves predictions of the scorer, even though the scorer does not consider the generated explanations during inference time. Furthermore, the coherent free-text explanations could also improve the interpretability of the predicted ratings (see examples in Table 7).

Rationalizer Sizes in COKE. In Table 2, we also show how scaling the size of the rationalizer helps improve Pearson's ρ. We note that our best-performing setup includes a T5 3B model as the rationalizer, along with the DeBERTa-V3-Small model as a scorer. It is interesting to note that COKE ends up being 2.18x better than GPT-3.5 in Table 1 while being approximately 58x smaller in parameter size as compared to it.

Varying $\mathcal{N}$ in COKE. In Figure 4, we also compare varying the number of candidate generations from $\mathcal{G}_{\theta_{\mathcal{R}}}$ while scoring an aspect-story pair. We see that increasing the number of generations, $\mathcal{N}$ improves the Pearson's Correlation Coefficient, thereby supporting our hypothesis that diversity of generations can help mimic various annotator preferences. Increasing $\mathcal{N}$ for COKE helps it surpass the human performance. We also note that increasing $\mathcal{N}$ is less costly as compared to LLM approaches shown in Table 1, because COKE uses a smaller, finetuned LM.

5 Applications of Keywords in COKE

The keyword rationales generated by COKE not only significantly improve the performance, but also being faithful because they are used as input for the scorer, similar to other faithful rationalization approaches like Jain et al. (2020). Moreover, the keywords provide more interpretable evaluation and more fine-grained evaluation based on user-provided keywords.

5.1 Human Evaluation for Considering User-provided Keywords

To support our results further, we conduct a small human evaluation experiment. For this task, we ask two annotators each to first read the story and the corresponding aspect and ask them to provide *one keyword or keyphrase* of their choice, along with a score that helps them to evaluate aspect-story-keyword triple (Appendix C.4). We conduct this experiment on a subset of 100 story-aspect pairs from our test set, with the help of annotators recruited via Amazon Mechanical Turk[2]. Here, we compare COKE with the No Rationalization baseline and find that COKE utilizes the keyword provided by the annotators and leads to an 29.2% relative improvement over the Pearson's Correlation Coefficient score. This validates that COKE

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

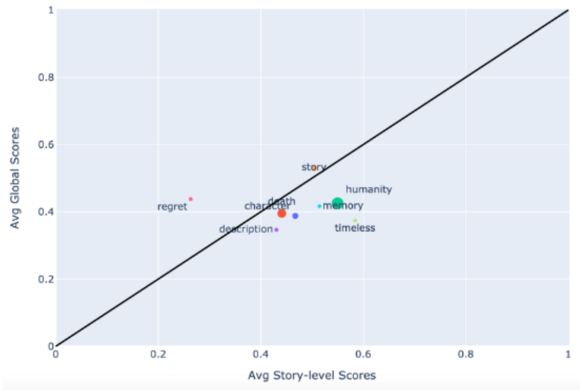

Figure 5: Suppose we want to understand the prediction rating of the *heartwarming/touch* aspect for a stroy, we can visualize the generated keywords in all of the generated samples. The x-axis plots the average rating of the keyword for this story, and the y-axis plots the global rating of the keyword averaged across the training set. The size of the keyword proportional to its frequency in the generated keyword sequences.

can better correlate with annotator-provided fine-grained keywords that baselines that do not have any keywords in them.

5.2 Keyword Visualizaion of CoKE

A scorer without the help of a rationalizer could only provide a rating prediction for each aspect and users often want to know where the rating comes from. The keywords in CoKE allow user to visualize what causes the final rating prediction. For instance, Figure 5 illustrates that *humanity* tends to be a negative keyword in the training data but being a positive keyword for the *heartwarming* aspect of this story, so the depiction of the *humanity* in this story increase its final *touching* rating.

6 Related Work

Due to the importance of automatic story evaluation, several types of approaches have been proposed. ROUGE (Lin, 2004), BERTScore (Zhang et al., 2019), BARTScore (Yuan et al., 2021), and CTC (Deng et al., 2021) compare the similarity between the generated text and the reference story. Although being effective in many other text generation tasks, higher similarity to the reference story is not necessarily a better story. Another type of evaluation method injects some noise into the human-written stories to create the low-quality stories and train a classifier to separate them. Examples include UNION (Guan and Huang, 2020), MANPLTS (Ghazarian et al., 2021), UNIEVAL (Zhong et al., 2022), and DELTAScore (Xie et al., 2023). Although these methods are good at discovering the incoherency from smaller language models, they cannot be used to evaluate a human-written story given a fine-grained aspect. Recently, researchers propose many general-purpose evaluation methods based on LLMs. For example, GPTScore (Fu et al., 2023) and G-Eval (Liu et al., 2023) directly prompt the LLM and several open-source models distill LLMs to reduce the evaluation cost (Gao et al., 2024). Li et al. (2024b,a) summarize these LLM-as-judge studies well. In these papers, GPT-4 usually demonstrates the best correlation with human judgments.

Methodologically, our method is related to the LLM rationale generation and Minimum Bayes Risk (MBR) decoding (Bertsch et al., 2023). Recent work in generating fluent free-text rationales has made use of two types of approaches - fine-tuning a small language model with gold human written rationales (Camburu et al., 2018; Narang et al., 2020; Wiegreffe et al., 2021) or zero-shot prompting LLMs to generate free-text rationales (Jung et al., 2022; Wei et al., 2023; Kojima et al., 2023; Li et al., 2023; Lightman et al., 2023). Some approaches also leverage few-shot training approaches with a handful of gold rationales (Marasović et al., 2022; Chen et al., 2023). Our method could also be viewed as a special case of MBR, which generally refers to the methods that merge multiple generated candidate answers to improve the output quality. Other special cases of MRB include self-consistency prompting (Wang et al., 2022), crowd sampling (Suzgun et al., 2023), complex CoT (Fu et al., 2022), and output ensembling (Martinez Lorenzo et al., 2023).

7 Conclusion

In this study, we look at a simple, yet efficient way to evaluate story-aspect pairs. We propose CoKE that samples multiple generated keyword sequences before explanations, and using the generated keywords to score an aspect-story pair. We posit that sampling helps us get diverse annotator ratings, and using keywords helps alleviate the objective mismatch between generating coherent explanations vs. usable explanations for downstream scoring. We show that that keywords not only improve the rating prediction performances, but also make the evaluation more interpretable and controllable.

Limitations

This work focuses on the fine-grain story evaluation task, which causes two limitations. First, we do not know if COKE could also improve CoT in the other applications that involve subjective human judgements. Second, our choice of evaluation dataset is limited and it is hard to know if COKE could bring similar improvements in other types of stories.

In Table 2, we show that increasing the sizes of rationalizer could lead to better performance, but we do not have resources to fine-tune the LMs that are larger than 3b. Furthermore, most of our experiments in this work, while still relevant, are done before early 2024, so we did not evaluate the performance of large reasoning models such as o1 or o3. Nevertheless, reasoning models are expensive and not optimized for such subjective tasks, so COKE should still be state-of-the-art method in fine-grained story evaluation, especially when we consider the inference cost.

Finally, there are some more complex LLM-as-judges approaches. For example, Verga et al. (2024) show that prompting multiple LLMs to discuss with each other improves the quality and reduces the cost of the evaluation task. However, we believe that the large performance gap between COKE and the off-the-shelf LLMs in Table 1 demonstrate the prompting LLMs without customizing/fine-tuning the LLMs is not very likely to achieve state-of-the-art results in subjective story evaluation tasks.

Ethical Statement and Broader Impact

When dealing with ambiguity in evaluation tasks, one of the most common methods is to collect more fine-grained annotations (Wu et al., 2024). However, our work shows that some story evaluation tasks are so subjective that only collecting fine-grained annotations is not sufficient.

The rising of the large reasoning models demonstrates the potential of LLMs given a high quality evaluation model. Nevertheless, no reliable reward model exists in more subjective tasks such as story evaluation. Our work could potentially provide some useful clues for solving the great challenge.

Finally, although customizing evaluation model is necessary in some applications, consistently targeting audience might intensify the problems of the filter bubbles (Spohr, 2017). For example, using COKE to filter the story submissions could reduce the manually reviewing cost and make reviewing much more submissions possible, but it could also intensify the selection biases in the dataset that trains the evaluation model.

8 Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor. We thank Anjali Narayan-Chen, Alessandra Cervone and Shereen Oraby for discussions during this project. We also thank the USC INK Lab and Xiang Ren for feedback on the draft and work. Additionally, we thank anonymous reviewers and ACs for their feedback on improving this work.

References

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew R Gormley. 2023. It's mbr all the way down: Modern generation techniques through the lens of minimum bayes risk. In *Proceedings of the Big Picture Workshop*, pages 108–122.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Hong Chen, Duc Vo, Hiroya Takamura, Yusuke Miyao, and Hideki Nakayama. 2022. StoryER: Automatic story evaluation via ranking, rating and reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1739–1753, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wei-Lin Chen, An-Zi Yen, Hen-Hsen Huang, Cheng-Kuang Wu, and Hsin-Hsi Chen. 2023. Zara: Improving few-shot self-rationalization for small language models. *Preprint*, arXiv:2305.07355.

Domenic Cicchetti. 1994. Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instrument in psychology. *Psychological Assessment*, 6:284–290.

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.

Ehsan Doostmohammadi, Oskar Holmström, and Marco Kuhlmann. 2024. How reliable are automatic evaluation methods for instruction-tuned llms? *arXiv preprint arXiv:2402.10770*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Le Bronnec Florian, Verine Alexandre, Negrevergne Benjamin, Chevaleyre Yann, and Allauzen Alexandre. 2024. Exploring precision and recall to assess the quality and diversity of llms. *arXiv preprint arXiv:2402.10693*.

Eibe Frank, Gordon W Paynter, Ian H Witten, Carl Gutwin, and Craig G Nevill-Manning. 1999. Domain-specific keyphrase extraction. jcai'99: Proceedings of the sixteenth international joint conference on artificial intelligence, 668-673.

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

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024. Llm-based nlg evaluation: Current status and challenges. *arXiv preprint arXiv:2402.01383*.

Sarik Ghazarian, Zixi Liu, SM Akash, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. 2021. Plot-guided adversarial example construction for evaluating open-domain story generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4334–4344.

Pawel Gmyrek, Christoph Lutz, and Gemma Newlands. 2024. A technological construction of society: Comparing gpt-4 and human respondents for occupational evaluation in the uk. *Available at SSRN 4700366*.

Jian Guan and Minlie Huang. 2020. Union: An unreferenced metric for evaluating open-ended story generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9157–9166.

Elisabeth Gülich and Uta M Quasthoff. 1986. Storytelling in conversation: cognitive and interactive aspects. *Poetics*, 15(1-2):217–241.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. 2020. Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, Online. Association for Computational Linguistics.

Shengyu Jia, Tao Meng, Jieyu Zhao, and Kai-Wei Chang. 2020. Mitigating gender bias amplification in distribution by posterior regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2936–2942, Online. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. *arXiv preprint arXiv:2205.11822*.

Seungone Kim, Se Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The CoT collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12685–12708, Singapore. Association for Computational Linguistics.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of RLHF on LLM generalisation and diversity. In *The Twelfth International Conference on Learning Representations*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024a. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. 2024b. Leveraging large language models for nlg evaluation: A survey. *arXiv preprint arXiv:2401.07103*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

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

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Ana Marasović, Iz Beltagy, Doug Downey, and Matthew E. Peters. 2022. Few-shot self-rationalization with natural language prompts. *Preprint*, arXiv:2111.08284.

Abelardo Carlos Martinez Lorenzo, Pere Lluís Huguet Cabot, Roberto Navigli, et al. 2023. Amrs assemble! learning to ensemble with autoregressive models for amr parsing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1595–1605.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20.

Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A Smith. 2021. Annotators with attitudes: How annotator beliefs and identities bias toxic language detection. *arXiv preprint arXiv:2111.07997*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Dominic Spohr. 2017. Fake news and ideological polarization: Filter bubbles and selective exposure on social media. *Business information review*, 34(3):150–160.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2023. Follow the wisdom of the crowd: Effective text generation via minimum bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4265–4293.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. *Preprint*, arXiv:1411.5726.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Danqing Wang, Kevin Yang, Hanlin Zhu, Xiaomeng Yang, Andrew Cohen, Lei Li, and Yuandong Tian. 2023a. Learning personalized story evaluation. *arXiv preprint arXiv:2310.03304*.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023b. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5558, Toronto, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. 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 Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari

Ostendorf, and Hannaneh Hajishirzi. 2024. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36.

Zhuohan Xie, Miao Li, Trevor Cohn, and Jey Han Lau. 2023. Deltascore: Evaluating story generation with differentiating perturbations. *arXiv preprint arXiv:2303.08991*.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038.

A Rating Prediction Diversity

To verify that CoKE could model/output more diverse rationales and ratings, we compare the standard deviation (SD) of the ratings predicted by different methods. For each story-aspect pair, we compute the SD of ratings (0-1 range) before averaging them into the final prediction.

In Table 1, the SD of **CoKE (T5-3B)** (0.513) is much larger than the SD of **Mistral-7B-Instruct CoT SC Mean** (0.289) and **GPT 3.5 CoT SC Mean** (0.33). In Table 2, the SD of CoKE (**T5 Small** $\rightarrow (\mathcal{K}_{\text{TextRank}}(\mathbf{e}), \mathbf{e}) + (\mathbf{s}, \mathbf{a}, \text{TextRank}(\mathbf{e}))$ $\rightarrow$ **DeBERTa-V3 Small**) is 0.511, which is also much larger than 0.310 from $(\mathbf{a}, \mathbf{e}) \rightarrow$ **DeBERTa-V3 Small** and 0.337 from $(\mathbf{s}, \mathbf{a}, \mathbf{e}) \rightarrow$ **DeBERTa-V3 Small**.

The experiment verify that keyword extraction indeed drastically improves the diversity of the predicted ratings and it also suggests that the models that has a larger Pearson's ρ usually also has a larger SD (i.e., rating diversity).

B StoryER Dataset Analysis

The StoryER dataset (Chen et al., 2022) extends the WritingPrompts (Fan et al., 2018) dataset, which consists of multiple writing prompts and corresponding human-written stories for those prompts, by adding ratings for ten *aspects* that are picked by the authors from a given list of fixed aspects, along with *comments* that justify the corresponding ratings given.

Each of these aspects aims to highlight a separate semantic or literal aspect of the story – for example, aspects can highlight the 'ending' or 'humour'-level of a story. This is done by multiple annotators for every writing prompt + story pair, however the number of annotators, and actual aspects (out of ten) that are annotated for a story can vary. Figure 6 and Figure 7 show the distribution of annotator provided ratings on the training set of the dataset. Table 3 and Table 5 provide additional details of StoryER.

Split	Train	Dev	Test
Number	17982	4496	5631

Table 3: **Dataset details**: Since StoryER does not contain a validation set, we use the train set to create it. We partition the train set by unique writing prompts and split it into a train and validation set based on it.

Aspect	Percentage
Ending	19.91%
Character Shaping	18.20%
Scene Description	14.81%
Middle/Twist/Flow	14.11%
Opening/Beginning	12.90%
Novel/Idea	9.90%
Funny/Hilarious/Laugh	4.08%
Horror/Scary	2.94%
Sad/Crying/Tear	1.62%
Heartwarming/Touch	1.48%

Table 4: **Percentage Distribution of Aspects in Training Set**: Given that not all aspects are annotated for all stories, there is an imbalance in the distribution of aspects.

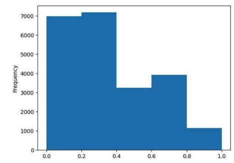

Figure 6: We plot the distribution of annotator provided ratings in the training set.

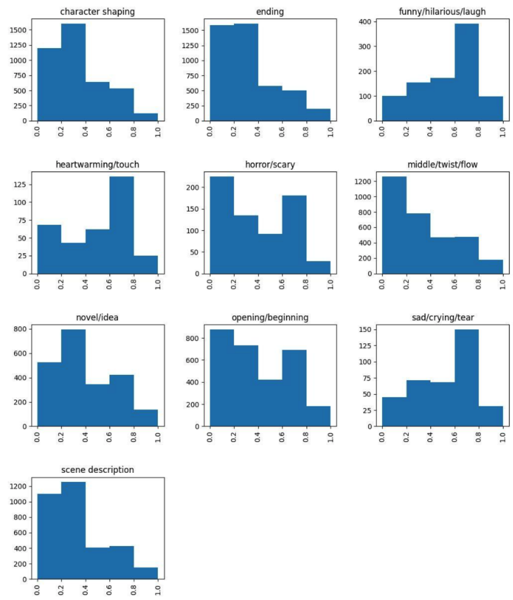

Figure 7: Distribution of annotator provided ratings across different aspects.

C CoKE Details

C.1 Training Parameters

For all the LLM generations (on GPT 3.5, 4, LLaMa, and Mistral), we set a temperature of 1 and maximum token length of 1024.

For training the rationalizer and scorer, we set

Writing Prompt	Story	Aspect, Score	Annotator Explanation
The cure for death was discovered and it worked 99% of the Earth's population. You are one of the 1% and now 90 years later, you are the last mortal left on your deathbed. The World comes to see the last dying human.	The world didn't mourn. It was a celebration. Confetti, streamers, loud fireworks while I laid quietly on my deathbed. Death was dead. Long live life. Or so they thought. They didn't understand what I understood. It wasn't because the cure didn't work on me, no, it was because I *didn't* want the cure. Bodies rot. Minds decay. Death is a mercy to rid the world of these ugly things. Death wasn't the problem, humanity is. In time, they would realize it. They would remember my name, the last mortal to die, and cry for the ability to do so. Unfortunately for them, Death will never come in time.	novel/idea, 5/5	The story was really written for its tiny size, it ended and gave a powerful message to the reach of the humanity, i bet for them first 100 or 200 years will be wonderful, i don't know how they will control the population though
You go to sleep on the night of your 25th birthday, only to wake up on your first day of 1st grade. You use your knowledge of the future to take advantage of the situation, and ball hard. However, when you come back to sleep the night of your 25th birthday, you wake up once again in 1st grade.	The clock ticks. I have one minute before I reach my silver year of life. I take this minute to reflect on my years. I was a very bratty child. I hated my teachers, as I thought that they were just other people in the world. I barely passed high school and took a couple weeks of college before I realized it wasn't what I was looking for in life. Since then, I had taken over my father's business in selling pools and spas as well as contracting. It was not a job I enjoyed, but it was one I had to do for my rent situation. 3...2...1... 11:42 PM on my birthday had passed. It was this day 25 years ago I had come out my mother's womb. Another year of a life that was just wasted. I had gone to sleep after this minute. Despite the momentous occasion, I still had a job to do early in the morning, and this customer was a particularly angry one. When I wake up, it is not the queen bed I have in my apartment, but the house I spent my early childhood in. Instead of the tall 6'3" body I had as an adult, I had the small body of a child. I look near my bed and see a face I had nearly forgotten. It was my old dog, Luna. She was already old when I was born and we were forced to put her down when I was merely 7 years old. I look at the calendar near my bed. It was about 19 years ago. I was 6 years old, about to go back to my first day of first grade. I realize something. First grade is when I changed from a curious child to a bratty child. Perhaps a higher power has sent me to fox my mistakes I have made. As I walk into class, I see many faces I had not seen in years. I look at my "beat friend" at this age, who grew up to be a crackhead. I look at my actual best friend, who looked just as snobbish as she described herself to be. Going home each day, I actually do my homework. I don't pay as much attention in class, as I had already learned this all in my old life. Over the years, I start making smarter decisions. Instead of joining a basketball league as a youth, I dedicate my time to writing stories, a dream I had in my teenagehood. The teachers view me as a prodigy who knows well past my age. I skip the 3rd grade due to my knowledge, but no more. Despite everything, I wasn't a prodigy in my past life, so I wasn't seen as the next Einstein. As I reach the puberty stages and a few years past that, I start attempting to care more for my body. Instead of having a mop for a head, I style my hair each day. By the time I am 15, I have a relationship with a friend from my past life, a stable one. Now, as I wait for my second 25th birthday, I sit back and realize that my life has changed. I managed to make my life better, but I cared little for others. Could I have done better. Of course I could. Would I want to start all over, of course not! 3... 2... 1... 11:42 has passed. I go to sleep. When I wake up, my body is once again too small and I wake up in an all to familiar bed. "It appears..." I whisper, "That you aren't done with me, yet.".	character shaping, 2/5	The author of this story was really unable to bring life to the identities and persona's of the characters in this story. Also they were no lively interactions between the characters.
In the future criminals are thrown into a forest completely surrounded by city. Civilians hunt them in the forest. Police watch the forest edge for criminals, and kill them if seen leaving. You were falsely accused of murder and thrown into the forest with 4 other criminals.	They left us deep in the woods with nothing but our orange jumpsuits, our handcuffs, and each other. Fifteen minutes, they had told us. Fifteen minutes and the handcuffs would open. Fifteen minutes and the gates would open, letting the hunters in. The others were talking. I ignored them. They were criminals, murderers. I was innocent. I looked at my handcuffs. I knew how they worked. Each cuff had a tracking chip. When they sent the signal that opened the gates, the cuffs opened too. That was good information to have. I rubbed my sternum. It was still sore. There was a tracking chip in me too, inside the bone. It tracked my position and heart rate. When I died, they would know it. If I tried to leave, they would see me. That was good information to have. One of the others, Dan, was too loud. He broke my train of thought. I had to think. There was a way out, but I had to think. "I won't be hunted! I won't! Not like some, some animal!" he shouted. "Some of them use dogs, you know! Better to just die now. If I make it to the edge, the guards will just shoot me. Better that way." He was rambling. He was frantic, manic. "The edge is too far," I said. "You won't make it before they let the hunters in." "Yeah, and what do you know? I heard you killed some kid. I done a lot of things, but I ain't never murdered no kid." He kept going. I ignored him. I hadn't killed anyone, at least not on purpose. "Shut up, both of you," said Fat Mike. We called him Big Mike to his face. "We need to get ready. Need to make weapons," said Fat Mike. "You want to fight guns with sticks?" Thin Mike scoffed. He was right. Fat Mike was right too. They were coming to kill us. It was kill or be killed out here. I hadn't killed anyone, at least not on purpose. I had to think. "Hey, where's Steve?" Fat Mike asked suddenly. I had noticed him slip off while the others were arguing, but I didn't say anything. "He stole my idea!" proclaimed Dan. "He's headed to the edge. A man shouldn't be hunted. Better that way." "I already told you, it's too far," I said. "Shove it," Dan replied angrily. "Might as well try." He turned his back to me. I slipped my hands over his head. The chain of my handcuffs pressed against his throat. I pulled as hard as I could. He struggled. "Better this way," I said. He struggled harder. I pulled harder. He stopped struggling. I let him fall. It had been easier than I thought it would be. That was good information to have. The Mikes were quiet. I ignored them. My cuffs unlocked. I let them fall. They were coming to kill us. It was kill or be killed out here. I hadn't killed anyone, at least no one who wasn't asking for it. I had to think.	scene description, 4/5	So actually the protagonist actually committed a crime and is not innocent, at least that's what was implied here "I hadn't killed anyone, at least not on purpose."

Table 5: **StoryER Dataset:** We give some examples of how StoryER stories and aspects, as well as human annotator explanations look like.

the parameters as shown in Table 6. The best checkpoints are chosen based on the lowest validation loss.

Config	Assignment
train batch size	4
eval batch size	4
seed	0
max epochs	25
learning rate	3e-5
learning scheduler	fixed
GPU	Quadro RTX 6000
Training time	4 hours

Table 6: **Training Parameters**: Here we show the models we used and hyperparameters we used training.

C.2 Human Performance Calculation

We then calculate different variants of human performance that is estimated from the multiple-annotator annotations that the StoryER test set contains. Figure 8 contains a visual description of these variants. Optimal Prediction and Majority Voting includes taking the mean and mode of the annotator predictions respectively as predictions. However, they work under the assumption that ratings of all annotators are available at test time, which is not a realistic setting. The Human Predicting Human variant randomly selects a rating from one annotator, and uses that as a prediction to estimate other annotators, which better represents the setting that our evaluation systems would fall into (assume the prediction from the system to be one 'annotator' that tries to best approximate other annotators).

C.3 Details about $\mathcal{K}(\cdot)$

For all of the keyword extractor methods, we set number of ngrams to be between one and three, so as to get a both keyword and keyphrases from the annotator explanations. We extract the top ten keywords produced by these extractor.

C.4 Human Evaluation

All our crowdworkers are from countries where English is the primary language. For all our human studies, the task is setup in a manner that ensure that the annotators receive compensation that is above minimum wage. Turkers were also chosen using extensive qualifications, where they had prior story reading and rating experience. We provide the task shown to turkers in Figures 9 and 10.

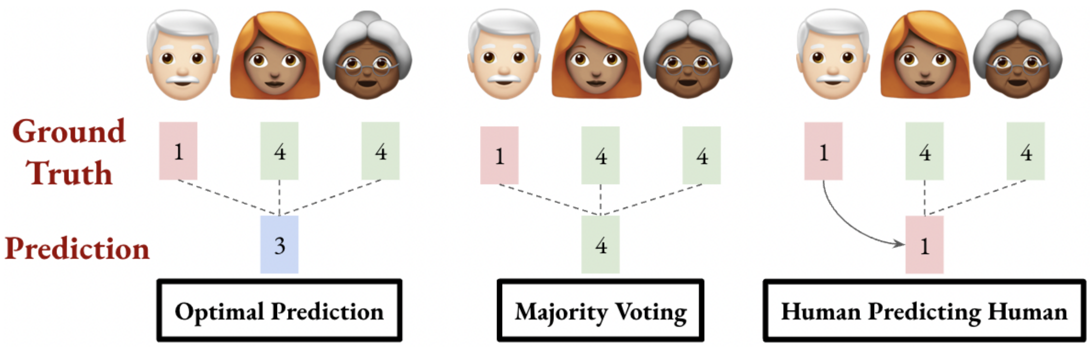

Figure 8: Different ways to calculate human performance - we use the *human predicting human* setting.

Aspect	Annotator Explanation	TextRank Keyword	Generated Keywords	Generated Explanation
character shaping	The author didn't do a good job to portray their characters in this story. The author should've at least detailed his main character a little bit better.	'little bit', 'main character', 'characters', 'story', 'author'	naive cliche,main character,diverse characters,personalities,names,family,conversation	The two diverse characters were nicely written, their conversation with their families wasn't cliche and had personalities all their own, they didn't stand out in the crowd as much.
heartwarming/touch	I would figure no matter the outcome when the kid came through the portal, even if your worst nightmare came out youd at least be cordial and make an attempt to be civil, not immediately come out swinging with the insults.	'worst nightmare', 'insults', 'attempt', 'kid', 'outcome', 'matter'	tame story,way,son,wife,decision,mom,man	I think this is a tame story because the man's decision to move in with his wife and son is pretty sweet. But the way he relates this is too shallow.
ending	The ending didn't make any sense at all, the story was too boring and bland for my taste, i was keeping my wits together just to complete reading this story	wits,taste,story,sense,ending	toon science,story,divots,detailing,ending	The ending was kind of weird. I was expecting something about fixing the divots but there was no detailing or even detailing in the story.

Table 7: **Example Generations:** We give some examples of how StoryER annotator explanations and extracted keywords look, along with generated keywords and explanations.

Instructions (click to collapse)

In this HIT, you will read a story. Then, you will be shown 3 *aspects* with respect to which you will have to rate the story. The aspects correspond to certain semantic story-specific components, like *character shaping*, *scene description*, *beginning or ending*, *flow*, etc to name a few. You will have to do the following task:

- **Provide <u>your own keyword</u> (a single word or a phrase) and rate the story according to the keyword:** *You will write your own keyword that focuses on certain parts of the story. You will then rate the story, with a focus on the keyword you wrote.*

HIT Details

Rating Scale

A rough reference of how you should rate the story is given below -

- **<u>1 (Very Poor)</u>:** *The story very poorly represents the aspect.*
- **<u>2 (Poor)</u>:** *The story poorly represents the aspect.*
- **<u>3 (Acceptable)</u>:** *The aspect is acceptable for the story.*
- **<u>4 (Good)</u>:** *The aspect is nicely represented in the story.*
- **<u>5 (Very Good)</u>:** *The story has a very good representation of the aspect.*

Keyword

In this task, we refer to keywords as names of characters, adjectives with/without adverbs decribing the aspect of the story, or important words within or about the story that help highlight the given aspect in the story. When you are asked to write your own keywords for the given story, think about words that can help describe how the aspect is represented in the story.

Example HIT

Story (Shown): *Here, we show a story about Jack who wants to kill an old man. Before the killing, Jack give a long and philosophic speech that makes him look wise. Later, the old man points out that the speech just shows that Jack is just too scared to actually kill him.*
Aspect (Shown): *character shaping*
Keyword we provide (Response): *wise*
Score (Response): *4.2*
Justification (Response): *Jack emphasizes on the fact that he is actually scared to kill the old man, which strengthens his character, and makes him wise.*

Figure 9: Instructions provided to turkers.

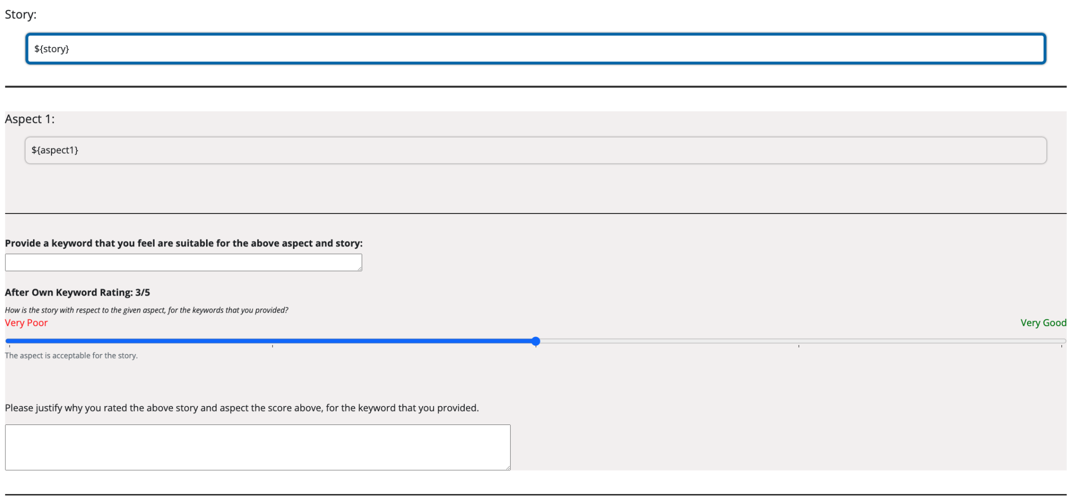

Figure 10: Actual task given to turkers.