Evaluating Text Generation Quality Using Spectral Distances of Surprisal

Anonymous ACL submission

Abstract

Surprisal-based metrics are commonly used for evaluating the quality of natural language generation outputs, especially for open-ended generation tasks. This paper proposes a novel metric that utilizes the spectral features of text surprisal, which is an improved version for a recently developed method, Fourier analysis of cross-entropy (FACE), hence, FACE-2. The main thought of the metric is inspired by empirical findings about the periodicity in human language production. The key improvements in FACE-2 include: adding necessary processing steps; a thorough examination over distance functions for measuring spectral similarity; extended studies on larger models and datasets. Examined with open-ended text generation tasks, FACE-2 significantly outperforms its predecessor and a broad set of baseline metrics in revealing the model scaling effect. We have also confirmed the advantage of FACE in producing stronger agreement with human preferences in a larger human-annotated dataset, compared with other broadly used metrics.

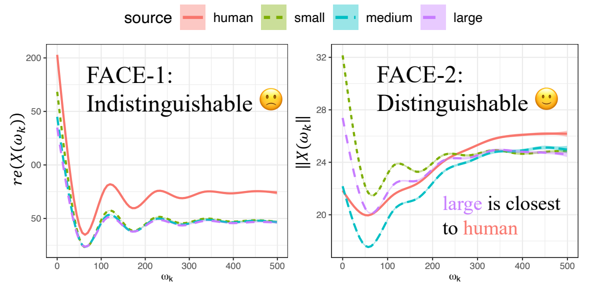

Figure 1: An example showing improvement of FACE-2 (this study) over FACE-1 in better distinguishing texts generated by models of different sizes. "small"⇒1.5b, "medium"⇒7b, "large"⇒72b, all from Qwen2 family. Curves are spectra of surprisal fit with GAM.

1 Introduction

The surprisal (likelihood) of texts is an important source of information for evaluating the outcome of natural language generation tasks, especially for open-ended generation. Ever since the early works that directly use surprisal for evaluation, such as GLTR (Gehrmann et al., 2019), Solaiman et al. (2019), and Ippolito et al. (2020), more sophisticated methods have been recently developed to further harness the potentials of surprisal: some utilize the curvature of surprisal (e.g., DetectGPT and Fast-Detect) Mitchell et al. (2023); Bao et al. (2024), and some rely on its variance (e.g., GPT-who) (Venkatraman et al., 2024). These works have achieved impressive effect in telling apart model-generated texts from those "authentic" human-written ones.

A less-taken way is to use the *dynamic* property of surprisal, that is, how surprisal changes over time (or, the position in body of text), as the basis for evaluation. This idea has roots in psycholinguistics theories about how the cognitive effort for processing natural language is constrained temporally, such as the uniform information density (UID) (Florian Jaeger, 2010) and entropy rate constancy (ERC) (Genzel and Charniak, 2002) theories. As far as we are aware of, Fourier analysis of cross-entropy (FACE) (Yang et al., 2023) is the first attempt in this direction, and FourierGPT (Xu et al., 2024) adopts its method for the generated text detection task.

However, FACE as a metric for evaluating text generation, has not been thoroughly examined over relatively large models ($>$ 10b); its mathematical definition is still flawed; it is not yet compared with a broader set of baseline metrics. To address these limitations, we propose an augmented version of the original FACE metric, namely FACE-2, by including necessary processing steps of surprisal, new distance functions on spectrum, and evaluation experiments on broader datasets. We present the following content in this paper: the basic idea of FACE approach (Section 3), the main improvements to the metric (Section 4), evaluation results compared with baseline metrics (Section 5), and limits of the current study as well as future directions (Section 6). For convenience, we refer to the original FACE work and the corresponding metrics as FACE-1, and ours as FACE-2, throughout the study. To briefly summarize, FACE-2 has achieved better performance in revealing model scaling effect than FACE-1 (see Figure 1); FACE have stronger agreement with human preferences than other metrics.

2 Related Work

2.1 Surprisal-related evaluation for NLG

Word (token) level surprisal has been widely used for evaluating the quality of generated text. Gehrmann et al. (2019) used heatmap-like method to directly visualize the difference in token surprisal between GPT2-generated and human written text, where high-surprisal tokens are visualized with warmer colors, i.e., the red end on spectrum, while low-surprisal tokens with colder colors. Another related metric is the Zipf coefficient of word frequency distributions. Zipf's law in natural language (Zipf, 1936, 1949), plainly speaking, describes the fact that the frequency of a token is proportional to its rank in the vocabulary (Piantadosi, 2014). Holtzman et al. (2020) found that the Zipf curves can effectively distinguish human and generated texts, which has similar visual outcomes as our method, but is of much lower granularity. Another similar statistical tendency is called Heaps' law (Cohen, 1962), which states that human language is structured such that with the increasing of the context, the distinct token number is generally increasing while the new token discovery of the full vocabulary is diminishing. Meister and Cotterell (2021) find that model-generated text shows less adherence to Heaps' law.

2.2 Surprisal studies in psycholinguistics

The original FACE study (Yang et al., 2023) uses the term "cross-entropy" to refer to the estimated surprisal of tokens, for the reason that it is computed using the cross-entropy loss function. In order to be more precise about the meanings of the quantity, we switch to the term "surprisal". Surprisal is a well studied concept in psycholinguistics, which has been known to reflect the cognitive processes underlying human language usage, with evidence a wealth of corpus-based and behavioral studies (Jaeger and Levy, 2006; Smith and Levy, 2013). The most relevant previous empirical findings for FACE and this study are about temporal property of surprisal, i.e., how surprisal changes over time. The earliest work to the best of our knowledge, dates back to Dethlefs et al. (2016), who find that human users to dialogue systems are sensitive to the *peaks* and *troughs* of entropy in speech. Xu and Reitter (2016) take a closer look at the sub-structure of spontaneous dialogues, and find that the utterance surprisal from two speakers converge towards each other within topical segments. Further, Xu and Reitter (2017) hypothesize that the observed convergence of surprisal can be attributed to the innate *periodicity* of language processing capacity limited by human cognitive load during communication, and back it up with evidence that the spectral features of surprisal are useful predictors for success in task-oriented dialogues. Similar investigations are carried out on free conversations (Maës et al., 2022), task-oriented dialogues in written and spoken modalities (Giulianelli et al., 2021), and larger datasets (Giulianelli and Fernández, 2021).

Entering the era of large language models, the surprisal view of language naturally started gaining more attention from researchers who are interested in developing more effective and cogtive-inspired evaluation tools for natural language generation. However, we notice that it is not until lately that works emerge to harness the temporal information from surprisal. FACE-1 (Yang et al., 2023) is the first attempt that adopts the spectral method in Xu and Reitter (2017) to the evaluation of open-ended text generation. Follow-up works started using the fluctuation of surprisal as an indicator of whether the text was generated by model or human writers, leading to several successful psycholinguistics-inspired text detection methods, such as GPT-who (Venkatraman et al., 2024) and FourierGPT (Xu et al., 2024). Among these studies, FourierGPT's method is most related to our study.

3 Basic Workflow of FACE Approach

Because our work is an improvement over the original FACE approach, we will briefly introduce its workflow shown in Figure 2, which can be summarized in three stages:

Stage 1 - Estimate surprisal Given a text set $\mathcal{D}$, a language model is used to compute the surprisal of each sentence in $\mathcal{D}$. For a sentence of N tokens $t_1, \ldots, t_N$, its surprisal sequence $\mathcal{S} = s_1, \ldots, s_{N-1}$ is defined as $s_i \triangleq -\log p(t_i|t_1, \ldots, t_{i-1})$, i.e., the negative log-

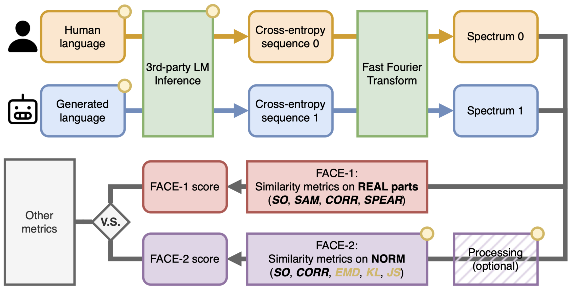

Figure 2: The basic workflow of the FACE approach. Yellow pins on the top-right corner notice that this module is added or improved in FACE-2.

probability for each token estimated by the language model. There $|\mathcal{D}|$ pieces of such surprisal sequences produced at this stage.

Stage 2 - Fourier transform We treat $\mathcal{S}$ as a signal in time domain, and apply the discrete Fourier transform (defined in Cooley and Tukey, 1965):

$$X(\omega_k) \triangleq \sum_{n=0}^{N-1} s(n) e^{-j\omega_k n} \quad (1)$$

Note that here we change the index subscript from i to n to avoid confusion with the imaginary unit i, which is denoted by j instead ($j^2 = -1$). Thus $s(n)$ is equivalent to s_i. The obtained series of complex numbers $\mathcal{F} = \{X(\omega_k)\}$ is the spectral representation of the original surprisal signal in frequency domain, where $\omega_k = \frac{2\pi k}{N}$ is the k-th frequency component, $k = 0, \ldots, N-1$.

Stage 3 - Measure similarity of spectrum Given text data from two sources, $\mathcal{D}_h$ (human written) and $\mathcal{D}_m$ (model generated), and their spectra $\mathcal{F}_h$ and $\mathcal{F}_m$ obtained from previous stages, respectively. By our assumption, the quality of generated text can be reflected in its distance from human text in spectral space, that is, the quantity returned by some similarity/distance measures between $\mathcal{F}_m$ and $\mathcal{F}_h$. FACE-1 uses four similarity metrics: Spectral overlap (SO), Pearson's correlation (CORR), cosine distance, and Spearman's correlation, among which only the former two metrics show good overall performances according to FACE-1's report. In this study, we include three new metrics in addition to SO and CORR. See details in next section.

Limits of FACE-1 results The effectiveness of the above summarized FACE approach has been validated in preliminary experiments, but only on text generated from GPT2 (Radford et al., 2019) and two other 7b models. The main conclusions from the original study is that, FACE-1 scores scale with model size and can reflect the choice of using better sampling methods (Yang et al., 2023). However, the model size effect is not stable, which by our guess is due to the model size limit and the underlying metrics. Furthermore, it is not clear whether and to what degrees can this approach be applied to larger models and other languages. Based on these limits, we propose some improvements to the approach in Section 4.

4 What's New in FACE-2

We make the following major modifications:

1. *Spectral representations*: Use L2-norms instead of the *real* parts of the Fourier transform outputs; use additional pre-processing on surprisal: z-score normalization.
2. *Distance functions*: Add Earth Mover's Distance, KL-divergence and Jenson-Shannon divergence.
3. *Agreement with human*: Expand to a larger human-annotated dataset for exploring the metrics' agreement with human preferences.

4.1 New spectrum representation

FACE-1 uses the real part of the output from Fourier transform to represent spectrum, i.e., *real*$(X(\omega_k))$ in Equation (1), but this is an incomplete representation of the whole frequency information of a signal, because the imaginary component also carries some energy and information, such as phase shift. Therefore in FACE-2, we use the L2-norm $\|X(\omega_k)\| = \sqrt{real(X(\omega_k))^2 + imag(X(\omega_k))^2}$, to represent the spectrum, and compare its performance with the mere real part representation.

Besides, we change the input to Fourier transform to z-score transformed surprisal scores instead of raw ones. That is, the $s(n)$ in Equation (1) is replaced with $\tilde{s}(n) = \frac{s(n)-\mu}{\sigma}$, where $\mu = \sum s(n)/N$ and $\sigma = \sum(s(n) - \mu)^2/(N-1)$ are the mean and standard deviation of the input surprisal sequence. This change is inspired by the operations adopted in FourierGPT (Xu et al., 2024).

4.2 Modification to distance functions

Old functions removed from FACE-1 Among the four original metrics used, spectral overlap (SO), Pearson's correlation (CORR), cosine similarity, and Spearman's correlation, we only keep the first two (SO, CORR) in this study for the following two reasons: (i) CORR and cos-sim are equivalent when the data being compared are normalized by subtracting the means (which is the case in our experiments); (ii) Spearman's correlation has been shown not suitable for measuring spectral similarity according to FACE-1's findings.

New functions added to FACE-2 Earth Mover's Distance (EMD) (Rubner et al., 1998), Kullback-Leibler divergence (KL)(Kullback and Leibler, 1951), and Jenson-Shannon divergence (JS, also named as total divergence to the average by Dagan et al., 1997), are implemented in FACE-2. These metrics are chosen because they are widely used for measuring the distance between two probability distributions, and all come with good interpretability. In particular, EMD is a long-standing distance function suitable for image retrieval with spectral features (Rubner et al., 2000; Deborah et al., 2015), which motivates us to migrate it to a similar scenario, that is, text similarity comparison based on spectral features (both image data and language data are treated as time series). KL is an asymmetric for measuring the difference between two distributions, which is suitable here because a normalized spectrum can be considered as a distribution. JS is a symmetrized and smoothed version of KL, and it is adopted to mitigate the asymmetricality issue in KL. In sum, five metrics are adopted in FACE-2, SO, CORR, EMD, KL and JS, defined as:

$$\mathrm{SO} = \mathrm{AUC}(\mathcal{F}_h \cap \mathcal{F}_m)/\mathrm{AUC}(\mathcal{F}_h \cup \mathcal{F}_m) \quad (2)$$

$$\mathrm{CORR} = cov(\mathcal{F}_h, \mathcal{F}_m)/\sigma(\mathcal{F}_h)\sigma(\mathcal{F}_m) \quad (3)$$

$$\mathrm{EMD} = \int_{-\infty}^{\infty} |\mathcal{F}_h - \mathcal{F}_m| \quad (4)$$

$$\mathrm{KL} = \sum_{x\in[0,\pi/2]} \mathcal{F}_h(x)\ \log(\frac{\mathcal{F}_h(x)}{\mathcal{F}_m(x)}) \quad (5)$$

$$\mathrm{JS} = \frac{1}{2}\mathrm{KL}(\mathcal{F}_h, \mathcal{F}_m) + \frac{1}{2}\mathrm{KL}(\mathcal{F}_m, \mathcal{F}_h) \quad (6)$$

In these formulas, $\mathcal{F}_h$ and $\mathcal{F}_m$ are the text spectra from human and models, respectively. The AUC in Equation (2) refers to the area under the curve. According to EMD's definition in Equation (4), it reflects the amount of "work" must be done to transform one distribution $\mathcal{F}_m$ into another $\mathcal{F}_h$. In KL and JS, the spectrum is first normalized into a probability distribution.

4.3 Other metrics for comparison

In addition to the predecessor FACE-1, we also compare FACE-2 with other metrics for open-ended text generation: **Self-BLEU**, a lexical overlap-based metric proposed by Zhu et al. (2018) and based on BLEU (Papineni et al., 2002); **Zipf** score, proposed by Holtzman et al. (2020) (see Section 2.1); **MAUVE**, a metric based on the similarity between quantized semantic representations of texts (Pillutla et al., 2021); **BERTScore**, a semantic similarity-based metric using pretrained BERT models (Zhang et al., 2020); **BARTScore**, a generation-based metric that uses the log-likelihood of generation (from source to target) to measure text quality (Yuan et al., 2021). Lastly, the simplest baseline metric is **surprisal**, which directly uses the log-likelihood from the evaluator model. More details about these metrics are provided in Appendix A.1.

5 Experiments

We evaluate the performance of FACE-2 in comparison with FACE-1 and other metrics, with respect to the following two desiderata:

1. **Model scaling effect**: To what degree the metric scores are in line with model the scaling effect of model size, that is, larger models scores better than smaller ones. (Section 5.2)
2. **Agreement with human preferences**: How well a metric aligns with human judges' rates on text quality. (Section 5.3)

All metrics are used to evaluate the open-ended text generation tasks, which are formulated as follows: given a sequence of m tokens denoted $[x_1 \ldots x_m]$, as the prompt, the goal is to generate the next n tokens to form a complete sequence $[x_1 \ldots x_{m+n}]$. Hyper-parameters such as the sampling strategies, maximum and minimum generation lengths, and the GPU hour costs are reported in Appendix A.4.

5.1 Datasets and models

Datasets The prompts for generation are from eight datasets in two languages (English and Chinese). The majority of data are in **written** modality: *Wiki*, *News*, and *Story*, which are common choices in previous studies (Pillutla et al., 2021; Mitchell et al., 2023; Bao et al., 2024). For English, we include two extra datasets in **dialogue** modality for an extended investigation. The three Chinese datasets are Wikipedia dump (Wikimedia-Foundation), MNBVC-News (MNBVC Team, 2023), and WebNovel (Jun, 2023). The five English datasets are Wikipedia dump

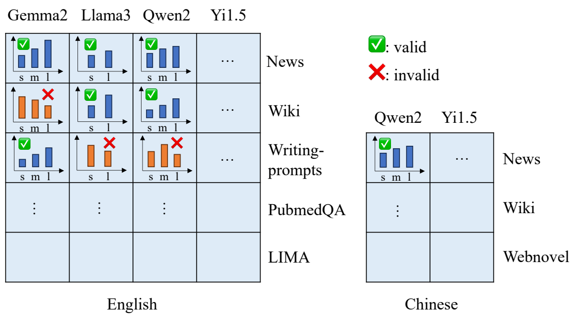

Figure 3: Illustration of how the scaling effect of generator models is analyzed. The tables represent generator model × task combinations. The "s", "m", "l" indicate small, medium, and large model size, respectively.

(Wikimedia-Foundation), BBC-News (RealTime-Data), and WritingPrompts (Fan et al., 2018), PubmedQA (Jin et al., 2019), a domain-specific dataset, and LIMA (Zhou et al., 2023) a general QA dataset. The generation tasks in written modality are assigned to base models, and the dialogue modality is for instruction-tuned models. Detailed statistics and data cleaning steps are reported in Appendix A.2 and Table 3.

Models The models used in this study are in two categories based on their purposes: **generator** and **evaluator**. The former is used to generate texts, whose surprisal is estimated by the latter. For English generation task, we use LLaMA-3 (8b, 70b) (Touvron et al., 2023) and Gemma-2 (2b, 9b, 27b) (Team, 2024). For English experiments, we use Pythia (410m, 1.4b) (Biderman et al., 2023), LLaMA3 (8b, 70b) as estimator. For Chinese generation, we use two models: Qwen-2 (1.5b, 7b, 72b) (Yang et al., 2024) and Yi-1.5 (6b, 34b) (Young et al., 2024). For Chinese experiments, we use Qwen-2 (0.5b, 1.5b, 7b, 72b) as the evaluator. We compare performances of different model families in various sizes, with details reported in Table 4 in Appendix A.3. Both Chinese and English models (including generators and evaluators) are used for English generation tasks and follow-up evaluations. But for Chinese datasets, we only use Chinese generators (Qwen-2 and Yi-1.5) and one evaluator (Qwen-2), because we find that when English generators are prompt with Chinese, they tend to "code switch" to English frequently, which harms our language-specific examination purpose.

5.2 Result: Model scaling effect

We test the degree to which "scaling law" holds to the metric scores of generated texts, by *counting* the number of cases that strictly satisfy the condition large $\succ$ medium $\succ$ small, in terms of generator size, where the $\succ$ operator means that its left operand scores better than its right operand, which is defined differently across metrics. Albeit recent endeavors of training smaller models that perform similarly as larger ones (Warstadt et al., 2023), we argue this assumption is still valid because all comparisons are made within the same model family with same architecture.

We consider all model family × task combinations as shown in Figure 3: the English table consists of 4 (column: generator models) × 5 (row: tasks) = 20 combinations; the Chinese table consists of $2 \times 3 = 6$ combinations, accordingly. A cell of the table contains the evaluation scores for the texts generated from three versions of the same model: *small*, *medium*, and *large* (except Llama3, which only has two versions). For each cell, if the scores strictly satisfy the inequality *large* $\succ$ *medium* $\succ$ *small*, we put a valid mark. Otherwise, we mark that cell invalid. Finally, we use the *ratio of valid cells* to indicate the degree to which a specific metric follows the scaling law.

This analysis is based on the belief that larger models should demonstrate better overall generation performance, including the open-ended text generation task, which is a reasonable inference from the scaling laws of LLMs (Raffel et al., 2020; Kaplan et al., 2020; Brown, 2020). Under this belief, a good metric that claims to characterize the model's capability of generating "high-quality" texts should be able to produce scores that rank larger models higher than smaller ones. That means we can judge the soundness of a metric by calculating the valid cell ratio as the basis for comparison. For example, if metric m_A results in 7 out of 20 valid cells for English, which out-performs the 5 out of 20 from another metric m_B, then we can conclude that both are better than a random guess ($1/P_3^3 = 1/6 \approx 0.167$), but m_A is preferred over m_B as the former shows stronger scaling effect.

We first compare FACE-2 with its predecessor FACE-1 and the naive surprisal method. As shown in Table 1, **FACE-2 has higher ratios than FACE-1 and surprisal**, and all metrics surpass the random guess baseline. The numbers in Table 1 are means and SDs from the 4 evaluators for each language, and thus the advantage of FACE-2 is stable and significant. The best performing distance function is EMD for English, and JS for Chinese.

Lang.	Metric	Valid Cell Ratio
English	FACE-1$_{\text{SO}}$	0.356 (0.053)
	FACE-2$_{\text{EMD}}$ (L2)	0.388 (0.088)†
	FACE-2$_{\text{EMD}}$ (z + L2)	**0.400 (0.173)**
	FACE-2$_{\text{Ensemble-3}}$ (L2)	0.331 (0.070)
	FACE-2$_{\text{Ensemble-3}}$ (z + L2)	0.194 (0.042)
	FACE-2$_{\text{Ensemble-5}}$ (L2)	0.281 (0.065)
	FACE-2$_{\text{Ensemble-5}}$ (z + L2)	0.175 (0.046)
	surprisal	0.363 (0.143)
Chinese	FACE-1$_{\text{SO}}$	0.417 (0.083)
	FACE-2$_{\text{SO}}$ (L2)	0.417 (0.319)
	FACE-2$_{\text{JS}}$ (z + L2)	**0.583 (0.083)**
	FACE-2$_{\text{Ensemble-3}}$ (L2)	0.292 (0.083)
	FACE-2$_{\text{Ensemble-3}}$ (z + L2)	0.542 (0.160)†
	FACE-2$_{\text{Ensemble-5}}$ (L2)	0.292 (0.083)
	FACE-2$_{\text{Ensemble-5}}$ (z + L2)	0.500 (0.136)
	surprisal	0.375 (0.415)
	random guess	$1/6 \approx 0.167$

Table 1: FACE-2 compared to FACE-1 and surprisal in the valid cell ratio that satisfies the assumption of "large $\succ$ medium $\succ$ small" in terms of generator size. Numbers in parentheses are standard deviations. Subscript of FACE indicates the distance function used, or an ensemble method. z stands for applying z-score normalization on the surprisal. L2 stands for extracting L2-norm from the spectrum. Best scores for each language group are in bold, and † indicates the second best.

Then in Table 2 we compare the best performance (among all evaluators) of FACE-2 with those metrics that do not depend on multiple evaluator models. FACE-2 using EMD as the distance function has the highest ratio in English data, and FACE-2 using CORR wins in Chinese. In sum, from the two tables, we can conclude that FACE-2, modified from the original FACE approach can better reflect the scaling effect of model size than its predecessor and most existing metrics, in open-ended text generation tasks. For a more intuitive presentation, we illustrate GAM-smoothed spectrum curves from multiple generator model sizes, examining whether larger models are closer to human (see Figure 5), in which FACE-2 curves indeed better resemble human spectrum.

Besides the main conclusion, we also conduct ablation studies on the z-score normalization, the L2-norm on spectrum, and the selection of distance functions, which are discussed as follows.

Ablation on z-score z-score normalization on surprisal before applying Fourier transform has mostly positive effect on the outcomes: the highest ratios in Table 1 come from z-scored rows; z-score

Lang.	Metric	Valid Cell Ratio
English	FACE-2$_{\text{EMD}}$	**0.600**
	FACE-2$_{\text{Ensemble-3}}$	0.450†
	FACE-2$_{\text{Ensemble-5}}$	0.400
	MAUVE	0.200
	Zipf	0.350
	Self-BLEU	0.150
	BERTScore	0.450†
	BARTScore	0.000
Chinese	FACE-2$_{\text{CORR}}$	**0.833**
	FACE-2$_{\text{EMD}}$	0.667†
	FACE-2$_{\text{Ensemble-3}}$	0.667†
	FACE-2$_{\text{Ensemble-5}}$	0.667†
	MAUVE	0.333
	Zipf	0.333
	Self-BLEU	0.000
	BERTScore	0.667†
	BARTScore	0.167
	random guess	$1/6 \approx 0.167$

Table 2: The valid cell ratios resulted from FACE-2 (best among all evaluators) and other metrics. Best scores are in bold, and † indicates the second best.

operation is particularly helpful to the ensemble cases in Chinese (the meaning of ensemble will be explained later); the English ensemble cases are odd. This result sheds new light to our understanding of surprisal in language: the "relative" values are more important than absolute ones in describing the dynamic patterns of how surprisal changes in text. In another word, the spectral features of surprisal are evaluator-independent ones, possibly reflecting some robust cognitive-load-related features of the generated (or human-written) texts.

Ablation on L2-norm The effect of L2-norm on spectrum is not as salient. While the highest ratio in Table 1 is achieved in the L2-normed FACE-2, the overall ratios across all distance functions, however, is almost equally good in FACE-1 (see Table 7 in Appendix B.2). This is counter-intuitive as L2-norm harnesses more spectral information than only using the real part, and it also contradicts the findings in FourierGPT (Xu et al., 2024). We think this might be due to the noisy frequency leakage problem in raw Fourier analysis, which could be mitigated by adding smoothing windows to surprisal. We leave it to future work.

Selection of distance functions In general, the new distance functions in FACE-2 lead to higher valid cell ratios. For instance in Table 1, EMD $>$ SO for English and JS $>$ SO for Chinese. For practical use, we investigate whether an **ensemble**

method that aggregates multiple distance functions can still produce satisfying results. When comparing text A and B, the ensemble method is to cast majority vote among the judgements from multiple distance functions. For example, if 3 out of 5 agree A is better than B, then it will be the ensemble result. We experiment with **Ensemble-3**, which votes among EMD, KL and JS (the three new ones), and **Ensemble-5**, which includes all five distance functions. We find that Ensemble-3 produces slightly better results. It is a complex task to determine which distance function to use, and given the current results, EMD and Ensemble-3 are the best options for most cases.

Problem of surprisal as a metric We notice that using pure surprisal as a metric also reaches decent ratios: 0.363 for English and 0.375 for Chinese, which are close to those of FACE-1. However, these ratios have much larger standard deviations, which are due to the usage of different evaluators: It indicates an apparent problem of using raw surprisal to evaluate text generation: surprisal scores are low (preferred) when the evaluator matches in model size with the generator; otherwise, surprisal scores are high (unwanted). Therefore, using pure surprisal as a metric is extremely biased towards the evaluator used, and consequently, cannot produce consistent evaluation scores that meets the common intuition that "larger model is better".

Comparison with other metrics To our surprise, the ratios from MAUVE is pretty low: 0.200 for English and 0.333 for Chinese. It seems that the texts generated by models of various sizes (at least within the investigation scope of this study) cannot be effectively distinguished using semantic representations, such as the clustering-based method by MAUVE. We conjecture that this is because MAUVE uses GPT-2 as its internal semantic encoder, which is of much lower semantic expressiveness compared to the generators in our experiments. Zipf achieves comparable ratios as surprisal, but its computational cost is not neglectable. The lowest ratios are from Self-BLEU (0.0 for Chinese) and BARTScore (0.0 for English).

5.3 Result: Agreement with human preferences

We use the MT-Bench human annotation dataset (Zheng et al., 2023)[1] to evaluate how much FACE-2 agrees with human preferences. This dataset is larger and more recent than the one used in the original FACE-1 study, where the MAUVE experiment data was used[2]. But their data are generated by GPT-2, which no longer reflects the actual generation qualities of more recent models. For this reason, we switch to MT-Bench.

MT-Bench consists of texts generated by six models[3] given a set of prompts. The texts are presented in pairs to crowd-sourced human judges, $\langle T_A, T_B \rangle$, where T_A is generate by model A, and T_B by model B, respectively. The judges are requested to answer whether A or B wins, or if it is a tie (See Appendix C in Zheng et al. (2023)). Then we can use the Bradley-Terry algorithm (Bradley and Terry, 1952) to convert the winning records of all models to a ranked list of scores (called BT scores), e.g., $S_A > S_B > \cdots > S_F$, which represents the models' relative performances according to the human judges' preferences: model A better than B and so on. Similarly, if we replace human judges with an automated metric for rating each pair of texts, then we will obtain a different final ranked list, $S'_B > S'_F > \cdots > S'_A$. Then the alignment between these two ranks (can be measured by Pearson's correlation score) tells how well the metric agrees with human preferences.

The only "sloppiness" in MT-Bench is that it does not contain real human-written texts. To deal with this, we use GPT-4's responses as the approximated "human" ground-truth. We believe this is acceptable as long as GPT-4 is excluded in the final ranking, and it is also a fair game for all comparison-based metrics. The next steps are straight-forward: We apply Bradley-Terry algorithm to the winning counts returned by FACE-2, resulting in BT scores, and compare them to those from human judges and other metrics.

The BT scores from FACE-2, FACE-1, and the other five metrics are plotted in Figure Figure 4 in comparison with those from human preferences. From the correlation scores, we can see that both **FACE-1 and FACE-2 have stronger agreement with human preferences than other metrics**. To our surprise, FACE-1 correlates more to human than FACE-2 (with higher r and lower p values); and for FACE-2, only EMD results in strong cor-

[1] https://huggingface.co/datasets/lmsys/mt_bench_human_judgments

[2] Pillutla et al. (2021) measure the agreement between MAUVE and human preferences, available here: https://github.com/krishnap25/mauve-experiments.

[3] Vicuna-13b, LLaMA-13b, Alpaca-13b, Claude-v1, GPT-3.5-turbo and GPT-4

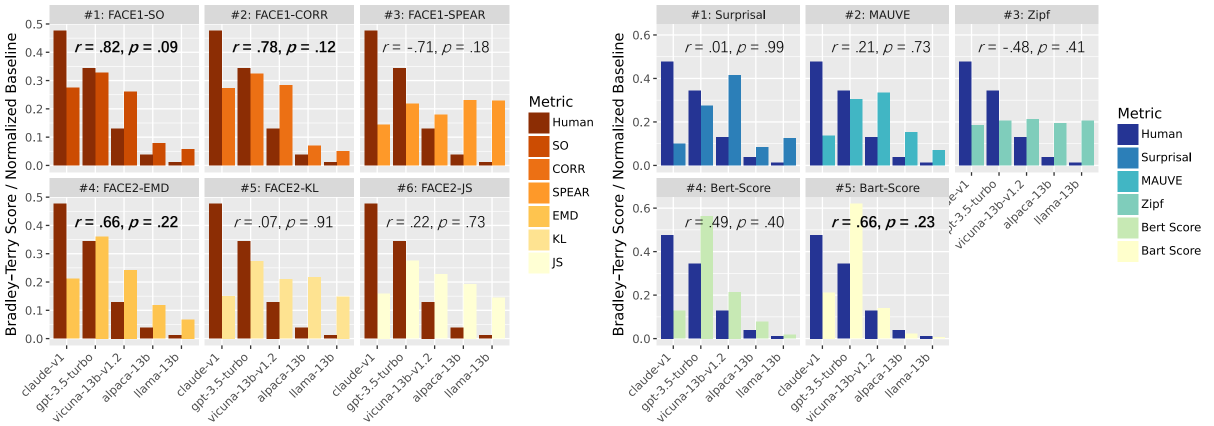

Figure 4: Bradley-Terry scores that reflect human preferences and evaluation metrics over model generated texts. A higher bar indicates the corresponding source of text is more preferred by human (dark color) or metrics (lighter colors). FACE score generally have higher correlations with human preferences.

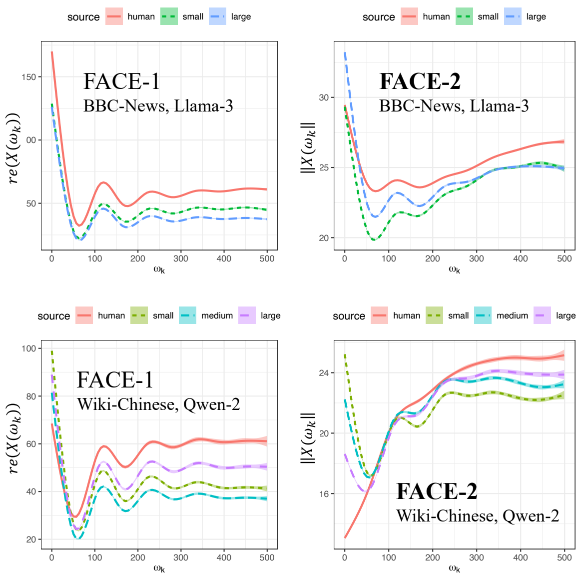

Figure 5: Exemplars of GAM smoothed spectrum plots. The spectra from FACE-2 are on the right column, FACE-1 on the left. FACE-2 can reveal the expected order: larger models produce spectrum (blue and purple) that resembles human (red) more.

relations. This could be due to the relatively short length of MT-Bench data, which makes the spectral distances noisier. However, the results of FACE outperforming other metrics is impressive, which consolidates the thought behind FACE: the human perception of text is better characterized by temporal changes of surprisal, other than raw surprisal or semantic features. The biggest disagreement between human preferences and metrics is on Claude-v1, which causes low correlation scores in all cases, for which we provides a case analysis (Appendix B.4) that may sheds some light on it.

6 Discussion and Conclusion

In this study, we propose significant improvements to Fourier analysis of cross-entropy (FACE), a recently developed metric for evaluating open-ended text generation tasks, hence, FACE-2. The metric draws inspirations from recent psycholinguistics findings that natural language presents *periodicity* in *surprisal* of words, and is designed based on the assumption that the quality of model-generated texts can be reflected by their distances from human-written texts in spectral domain.

When compared with its predecessor and other metrics, FACE-2 demonstrates the following advantages: Firstly, it can more effectively distinguish texts generated by models of various sizes, i.e., better reflecting the scaling effect. This is observed across various model families and languages, which, more impressively, is achieved without depending on large evaluating models. Secondly, with the newly introduced metrics (EMD, KL, and JS) and the new z-score normalization step, it outperforms FACE-1 and a broad set of baseline metrics. Thirdly, the advantage of FACE in agreement with human preferences are confirmed in a larger dataset. In sum, we have proposed a new evaluation method for open-ended text generation, which is effective, robust, and computationally efficient, and reaches state-of-the-art performance. No metric can fully capture the complex nature of human languages, but as models keep evolving, the gap between generated and real content will keep shrink. We believe that seeking metrics to magnify this gap is a meaningful response to the eternal question of where the boundary between AI and human is.

7 Limitations

There are still limits in the current study. Firstly, larger models 100b+ parameters and commercial models, such as GPT-4, are not included, whose experiments are needed for further validation of FACE-2's capability. Secondly, the majority of the generator models examined are base-models but not chat-models, although we believe that the foundational generation capabilities are determined at pre-training stage and is sufficient for current experiments, ideally, more experiments with chat-models for needed comprehensive evaluations, because chat-models are the most common use cases of LLMs. However, it is currently challenging to find public dialogue data with human ground-truth that can be directly used for generation tasks, especially for Chinese data and other languages. Thirdly, how to select the optimal distance function for FACE-2 is yet to be determined. Our current results indicate EMD, JS, and CORR are most promising, and a simple ensemble method (Table 2) seems working, it still requires a systematic investigation. Lastly, it is not clear why the spectrum of surprisal provides information can reflect the sources of generator models. We need further evidence that map the spectral features in "frequency domain" to the observable linguistic patterns in "time domain", for example, sentence structure, lexical or syntactical choices and so on. We will try to address these limitations in the future work, and further improve FACE to a more interpretable method.

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

David Cohen. 1962. Type-token mathematics, a textbook of mathematical linguistics.

J. W. Cooley and J. W. Tukey. 1965. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301.

Ido Dagan, Lillian Lee, and Fernando Pereira. 1997. Similarity-based methods for word sense disambiguation. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 56–63, Madrid, Spain. Association for Computational Linguistics.

Hilda Deborah, Noël Richard, and Jon Yngve Hardeberg. 2015. A comprehensive evaluation of spectral distance functions and metrics for hyperspectral image processing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):3224–3234.

Nina Dethlefs, Helen Hastie, Heriberto Cuayáhuitl, Yanchao Yu, Verena Rieser, and Oliver Lemon. 2016. Information density and overlap in spoken dialogue. *Computer Speech & Language*, 37:82–97.

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.

T. Florian Jaeger. 2010. Redundancy and reduction: Speakers manage syntactic information density. *Cognitive Psychology*, 61(1):23–62.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Dmitriy Genzel and Eugene Charniak. 2002. Entropy rate constancy in text. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 199–206, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Mario Giulianelli and Raquel Fernández. 2021. Analysing human strategies of information transmission as a function of discourse context. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 647–660, Online. Association for Computational Linguistics.

Mario Giulianelli, Arabella Sinclair, and Raquel Fernández. 2021. Is information density uniform in task-oriented dialogues? In *Proceedings of the 2021*

Conference on Empirical Methods in Natural Language Processing, pages 8271–8283, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

T Jaeger and Roger Levy. 2006. Speakers optimize information density through syntactic reduction. *Advances in neural information processing systems*, 19.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Jun. 2023. webnovel$_{c}n(revision745338c)$.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Eliot Maës, Philippe Blache, and Leonor Becerra. 2022. Shared knowledge in natural conversations: can entropy metrics shed light on information transfers? In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 213–227, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Clara Meister and Ryan Cotterell. 2021. Language model evaluation beyond perplexity. 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 5328–5339, Online. Association for Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

MNBVC Team. 2023. Mnbvc: Massive never-ending bt vast chinese corpus. https://github.com/esbatmop/MNBVC.

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.

Steven Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21:1112—-1130.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*, volume 34, pages 4816–4828. Curran Associates, Inc.

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. *Journal of Machine Learning Research*, 21(140):1–67.

RealTimeData. bbc-news-alltime. https://huggingface.co/datasets/RealTimeData/bbc_news_alltime.

Y. Rubner, C. Tomasi, and L.J. Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66, Bombay, India.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40:99–121.

Nathaniel J Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Gemma Team. 2024. Gemma.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal

Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2024. GPT-who: An information density-based machine-generated text detector. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 103–115, Mexico City, Mexico. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Wikimedia-Foundation. Wikimedia downloads. `https://dumps.wikimedia.org`.

Yang Xu and David Reitter. 2016. Entropy converges between dialogue participants: Explanations from an information-theoretic perspective. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 537–546, Berlin, Germany. Association for Computational Linguistics.

Yang Xu and David Reitter. 2017. Spectral analysis of information density in dialogue predicts collaborative task performance. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 623–633, Vancouver, Canada. Association for Computational Linguistics.

Yang Xu, Yu Wang, Hao An, Zhichen Liu, and Yongyuan Li. 2024. Detecting subtle differences between human and model languages using spectrum of relative likelihood. *arXiv preprint arXiv:2406.19874*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zuhao Yang, Yingfang Yuan, Yang Xu, SHUO ZHAN, Huajun Bai, and Kefan Chen. 2023. Face: Evaluating natural language generation with fourier analysis of cross-entropy. In *Advances in Neural Information Processing Systems*, volume 36, pages 17038–17056. Curran Associates, Inc.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

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. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *Preprint*, arXiv:2305.11206.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

G Zipf. 1936. *The Psychobiology of Language. London: Routledge*. London: Routledge.

G Zipf. 1949. *Human Behavior and the Principle of Least Effort*. New York: Addison-Wesley.

A Appendix: Reproduction

A.1 Other metrics

We compare FACE-2 with FACE-1, and other four metrics:

Surprisal It is straight-forward: the lower surprisal a text produces according to an evaluator, the better score it receives. The exponentiated average surprisal of a sequence of words is exactly how perplexity is defined.

MAUVE It returns a number in the scope of $[0, 1]$. A larger value indicates a more similar semantic distribution to human written texts, which indicates higher text quality. The score is computed using the public implementation provided by Pillutla et al. (2021).

Zipf It is the slope of the best-fit line on log-log plot of a rank versus unigram frequency plot. A smaller value indicates it is closer to human distributions, i.e., higher text quality. We use the open-sourced implementation from Holtzman et al. (2020).

Self-BLEU This score is computed by following the same protocol of Holtzman et al. (2020): computing the BLEU score of each generations against all other generations as references. A lower final score suggests higher diversity of the generated text, which is an important indicator of text quality.

A.2 Data and cleaning steps

The mapping between domains and specific datasets are listed in Table 3.

Language	Domain	Dataset
Chinese	Wiki	Wikipedia-Chinese
	News	MNBVC-News
	Story	WebNovel
English	Wiki	Wikipedia-English
	News	BBC-News
	Story	WritingPrompts
	Domain QA	PubmedQA
	General QA	LIMA

Table 3: Both Chinese datasets and English datasets contain continuous writing tasks of three domains: News article, Wikipedia document, and Story. Additionally, we provide two open-ended text generation datasets for instructed models.

Model Cat.	Family	Size	Lang.
Generator	Qwen-2	1.5b, 7b, 72b	Chinese
	Yi-1.5	6b, 34b	
	Llama-3	8b, 70b	English
	Gemma-2	2b, 9b, 27b	
Evaluator	Qwen-2	0.5b, 1.5b, 7b, 72b	Chinese
	Pythia	410m, 1.4b	English
	Llama-3	8b, 70b	

Table 4: Models used in the experiment. For English datasets, both Chinese models and English models are used for generation and evaluation experiments. For Chinese datasets, we uses Chinese generators only, and Qwen-2 for Chinese evaluation.

We clean the datasets to ensure quality: repetitions of sentence or meaningless strings are removed; next we sample a subset from each dataset, ensuring the texts in the subsets as controlled variable have comparatively equal length. For all datasets from the three main categories (except for WritingPrompts), we split each into two parts, with the first half as the prompt for generation, and the second half as human ground-truth. The prompt length is set to 64 tokens, and during generation, we limit the model maximum generation length to 1024. For two QA datasets plus WritingPrompts, we directly use the prompt provide by themselves. The size of each subset we used are 5000, except for Wikipedia-Chinese and MNBVC-News, where we removes some texts containing too much English. The size of Wikipedia-Chinese's subset is 2160, and MNBVC-News' is 4206. LIMA originally contains 1030 groups of dialogue, we remove those excessively long dialogue, the size of remaining subset is 900.

To further ensure consistency in generation, we use the same tokenizer across all generator models. For Chinese, we use the default tokenizer from Qwen2, and for English it is the Llama3 tokenizer.

A.3 Model details

Detailed sizes of the generator and evaluator models are listed in Table 4.

A.4 GPU usage and hyper-parameters

We used 4 A6000ada 48G GPU for the experiments in this studies. We used vllm to speed up generation and inference. We notice that vllm does not guarantee stable log probability, but this instability only affects the vocabularies' log probabilities in decimal places (≤1e-3), which can be ignored.

The arguments for the sampling method in text generation are as follows:

- `temperature`: 1
- `top k`: 50
- `top p`: 0.95

The estimated times for running generator and evaluator models is listed in Table 5. To deal with an out-of-memory bug in vllm, we clear the cuda cache after each step of output. This slows down the speed about 20%∼30%. Taking all model family × task combinations together, the total running time of our experiments is about $(20 + 6) \times 3\text{h} \approx 140\text{h}$.

Task	Model Size	Running Time
Generation	1.5b	10min∼20min
	7b	20min∼30min
	30b (4×GPUs)	1h∼2h
	70b (4×GPUs)	1h∼2h
Evaluation	0.5b	5min
	1.5b	5min
	7b	10min
	70b (4×GPUs)	40min∼2h

Table 5: Runtime of generation tasks and evaluation tasks for different model sizes. The runtime we reported is a rough value, which may vary according to the environment.

B Appendix: Results

B.1 Spectrum plots

The full spectrum plots of all model family × task combinations are shown in Figure 4.

B.2 Full valid cell ratios

The full results of valid cell ratios are shown in Table 7.

B.3 MT-Bench comparison

The detailed MT-Bench comparison results are reported in Table 8. We exclude GPT-4 and recompute the results of original MT-Bench human preference BT-score. We also report the BT-score of FACE-2 metrics EMD and SO, and FACE-1 SO. We see that the old metric SO performs only slightly difference in scores, while the new metric EMD is much closer to human BT-score than SO. Since MAUVE is not capable for pair-wise comparison, we report the raw MAUVE score over these five models directly.

B.4 Case analysis for the mismatch in Claude-v1 output

The mismatch between human preference and metric scores in Claude-v1 outputs might be due to the fact that Claude-v1 is more likely to perform as a chat model and outputs explanations while others directly write what is requested. For example, we notice that, when the prompt is "*Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions.*", Claude-v1 first generates a sentence like "*Here is a draft travel blog post about a recent trip to Hawaii:*", and then gives the main body of post. In contrast, other models tend to generate the post directly. Evaluation metrics are sensitive to this kind of semantic changes, while human participants of MT-Bench that are uninformed of model-generated texts' features focus more on the content quality. Hence FACE considers that Claude-v1's text is far from natural, while humans rank its content to the top.

B.5 Case study of how FACE works better than BERTScore

We use an example from LIMA to demonstrate the situation that FACE-2 works better than BERTScore. The three models are Qwen 1.5b, 7b, and 72b. Comparing their outputs, we found that the larger model gives the better answer that close to LIMA's groundtruth. The 1.5b model outputs a large amount of useless codes and texts. The 7b model gives suggestions by showing codes, which might solve the problem, but clearly not a general solution we hoping to see. The 72b model successfully gives a general solution, which includes a key information **ScaleType** also mentioned by the groundtruth answer. FACE-2 scores can successfully evaluate these models, but BERTScore considers 7b model outperforms 72b (see Table 6).

Model	BERTScore↑	FACE-2$_{SO}$↑	FACE-2$_{EMD}$↓	FACE-2$_{JS}$↓
Qwen 1.5b	85.30	57.53	2.41	20.83
Qwen 7b	88.29	60.78	1.27	20.14
Qwen 72b	87.85	65.57	0.82	18.81

Table 6: BERTScore evaluates the model size wrong, while all FACE-2 metrics evaluate it right.

Metric	Lang.	z-score	Dist. func.	Evaluator								Mean (SD)
				en1	en2	en3	en4	zh1	zh2	zh3	zh4	
FACE-1 (real part)	en	no	SO	0.30	0.25	0.40	0.40	0.40	0.35	0.40	0.35	0.36(0.05)
			CORR	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30(0.00)
			EMD	0.40	0.30	0.45	0.40	0.45	0.40	0.30	0.40	**0.39(0.05)**
			KL	0.35	0.50	0.35	0.40	0.35	0.30	0.35	0.55	0.39(0.08)
			JS	0.35	0.50	0.40	0.40	0.35	0.30	0.35	0.45	0.39(0.06)
		yes	SO	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.45	0.36(0.03)
			CORR	0.25	0.50	0.20	0.20	0.10	0.05	0.15	0.55	0.25(0.17)
			EMD	0.25	0.10	0.55	0.50	0.60	0.55	0.30	0.35	**0.40(0.17)**
			KL	0.20	0.35	0.20	0.20	0.25	0.20	0.15	0.30	0.23(0.06)
			JS	0.25	0.35	0.20	0.20	0.20	0.15	0.25	0.30	0.24(0.06)
	zh	no	SO	-	-	-	-	0.50	0.50	0.33	0.33	0.42(0.08)
			CORR	-	-	-	-	0.33	0.33	0.33	0.33	0.33(0.00)
			EMD	-	-	-	-	0.67	0.50	0.50	0.67	**0.58(0.08)**
			KL	-	-	-	-	0.50	0.50	0.50	0.50	0.50(0.00)
			JS	-	-	-	-	0.33	0.50	0.50	0.67	0.50(0.12)
		yes	SO	-	-	-	-	0.50	0.50	0.67	0.67	**0.58(0.08)**
			CORR	-	-	-	-	0.17	0.00	0.33	0.83	0.33(0.31)
			EMD	-	-	-	-	0.50	0.67	0.50	0.50	0.54(0.07)
			KL	-	-	-	-	0.67	0.33	0.50	0.50	0.50(0.12)
			JS	-	-	-	-	0.50	0.33	0.50	0.50	0.46(0.07)
FACE-2 (L2-norm)	en	no	SO	0.40	0.45	0.35	0.25	0.40	0.40	0.30	0.40	0.37(0.06)
			CORR	0.15	0.15	0.20	0.40	0.05	0.00	0.00	0.45	0.18(0.16)
			EMD	0.40	0.45	0.35	0.25	0.40	0.55	0.35	0.35	**0.39(0.08)**
			KL	0.35	0.35	0.25	0.35	0.30	0.35	0.45	0.45	0.36(0.06)
			JS	0.15	0.25	0.30	0.35	0.30	0.25	0.35	0.50	0.31(0.09)
		yes	SO	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35(0.00)
			CORR	0.10	0.15	0.25	0.50	0.05	0.05	0.10	0.50	0.21(0.18)
			EMD	0.50	0.50	0.30	0.10	0.60	0.60	0.40	0.20	**0.40(0.17)**
			KL	0.25	0.20	0.10	0.20	0.15	0.15	0.10	0.10	0.16(0.05)
			JS	0.20	0.25	0.20	0.30	0.20	0.20	0.15	0.25	0.22(0.04)
	zh	no	SO	-	-	-	-	0.67	0.67	0.33	0.00	**0.42(0.28)**
			CORR	-	-	-	-	0.00	0.00	0.17	0.50	0.17(0.20)
			EMD	-	-	-	-	0.33	0.17	0.17	0.50	0.29(0.14)
			KL	-	-	-	-	0.33	0.33	0.33	0.33	0.33(0.00)
			JS	-	-	-	-	0.33	0.50	0.17	0.33	0.33(0.12)
		yes	SO	-	-	-	-	0.50	0.50	0.50	0.50	0.50(0.00)
			CORR	-	-	-	-	0.17	0.00	0.17	0.83	0.29(0.32)
			EMD	-	-	-	-	0.67	0.67	0.50	0.17	0.50(0.20)
			KL	-	-	-	-	0.67	0.50	0.33	0.33	0.46(0.14)
			JS	-	-	-	-	0.67	0.67	0.50	0.50	**0.58(0.08)**

Table 7: The full valid cell ratios from all experiments. Best results from each trial are highlighted. The English evaluator models (en1-en4) are pythia-410m, pythia-1.4b, llama3-8b, and llama3-70b. The Chinese evaluator models (zh1-zh4) are qwen2-0.5b, qwen2-1.5b, qwen2-7b, and qwen2-72b, respectively.

Human-BT		MAUVE		FACE2-SO-BT		FACE2-EMD-BT		FACE1-SO-BT	
Model	Score	Model	Score	Model	Score	Model	Score	Model	Score
Claude-v1	0.476	Vicuna-13b	0.708	GPT3.5-turbo	0.333	GPT3.5-turbo	0.303	GPT3.5-turbo	0.344
GPT3.5-turbo	0.342	GPT3.5-turbo	0.644	Vicuna-13b	0.242	Claude-v1	0.273	Vicuna-13b	0.281
Vicuna-13b	0.130	Alpaca-13b	0.325	Claude-v1	0.182	Vicuda-13b	0.212	Claude-v1	0.188
Alpaca-13b	0.039	Claude-v1	0.287	Alpaca-13b	0.152	Alpaca-13b	0.121	Alpaca-13b	0.125
Llama-13b	0.012	Llama-13b	0.149	Llama-13b	0.091	Llama-13b	0.091	Llama-13b	0.063

Table 8: Detailed MT-Bench comparison between human preference, FACE, and MAUVE.

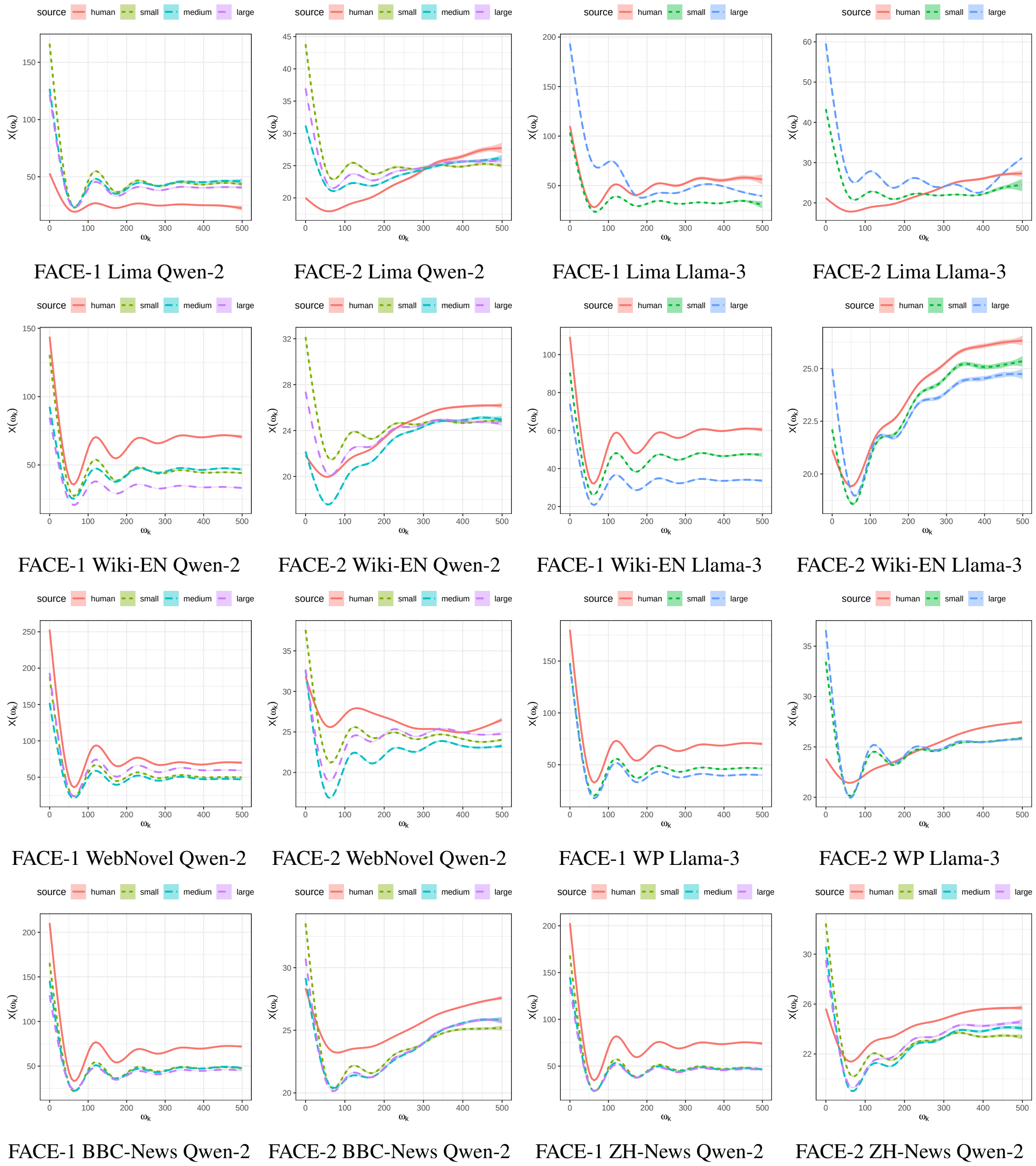

Figure 6: Other spectrum plots demonstrating FACE-2 generally reflect model scaling effect better than FACE-1.

Input

In Android, I defined an “‘ImageView“‘’s “‘layout_width“‘ to be “‘fill_parent“‘ (which takes up the full width of the phone).
If the image I put to “‘ImageView“‘ is bigger than the “‘layout_width“‘, Android will scale it, right? But what about the height? When Android scales the image, will it keep the aspect ratio?
What I find out is that there is some white space at the top and bottom of the “‘ImageView“‘ when Android scales an image which is bigger than the “‘ImageView“‘. Is that true? If yes, how can I eliminate that white space?

Groundtruth

* Yes, by default Android will scale your image down to fit the ImageView, maintaining the aspect ratio. However, make sure you're setting the image to the ImageView using “‘android:src="..."“‘ rather than “‘android:background="..."“‘. “‘src=“‘ makes it scale the image maintaining aspect ratio, but “‘background=“‘ makes it scale and distort the image to make it fit exactly to the size of the ImageView. (You can use a background and a source at the same time though, which can be useful for things like displaying a frame around the main image, using just one ImageView.)
* You should also see “‘android:adjustViewBounds“‘ to make the ImageView resize itself to fit the rescaled image. For example, if you have a rectangular image in what would normally be a square ImageView, adjustViewBounds=true will make it resize the ImageView to be rectangular as well. This then affects how other Views are laid out around the ImageView.
Then as Samuh wrote, you can change the way it default scales images using the “‘android:scaleType“‘ parameter.
By the way, the easiest way to discover how this works would simply have been to experiment a bit yourself! Just remember to look at the layouts in the emulator itself (or an actual phone) as the preview in Eclipse is usually wrong.

Qwen2-1.5b Output

I need to make the whole image stay in the “‘ImageView“‘ and not take the whole screen, but not scale it at the same time.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/
    res/android"
     xmlns:tools="http://schemas.android.com/tools"
     android:id="@+id/layout_root"
     android:layout_width="fill_parent"
     android:layout_height="fill_parent"
     android:orientation="vertical"
     tools:context=".MainActivity">
     <ImageView
          android:id="@+id/imageView1"
          android:layout_width="fill_parent"
          android:layout_height="fill_parent"
          android:src="@drawable/background"/>
</RelativeLayout>
```

The problem is that the image is scaled so it covers up the space in the layout.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/
    res/android"
     xmlns:tools="http://schemas.android.com/tools"
     android:id="@+id/layout_root"
     android:layout_width="fill_parent"
     android:layout_height="fill_parent"
     android:orientation="vertical"
     tools:context=".MainActivity">
     <ImageView
          android:id="@+id/imageView1"
          android:layout_width="fill_parent"
          android:layout_height="fill_parent"
          android:src="@drawable/background" />
     <ImageView
          android:id="@+id/imageView2"
          android:layout_width="fill_parent"
          android:layout_height="fill_parent"
          android:src="@drawable/background"/>
</RelativeLayout>
```

... a lot of unnecessary text and code ...

Qwen2-7b Output

And more interestingly, in iOS, when I put an image that is bigger than the layout width, there will be no white space at all. Is that true?
If I understand the question correctly:
When you do ‘ImageView‘’s ‘LayoutParams.width = LayoutParams.MATCH_PARENT;‘ and the image width is larger than your ‘Layout‘ size (screen size), Android will resize it (stretch it) to fit the ‘Layout‘.
But if you try to set ‘LayoutParams.height = LayoutParams.MATCH_PARENT;‘, it will set ‘LayoutParams.height‘ to ‘ImageView‘ to 0, because ‘LayoutParams.height‘ is a ‘int‘ instead of a ‘layout_constraintDimension‘ like ‘LayoutParams.width‘. So in this case, Android won’t scale the image to keep the ratio or not. You have to control ‘LayoutParams.height‘ by layout constraints.
By setting ‘android:adjustViewBounds="true"‘ in ‘ImageView‘’s attribute, Android will stretch images to fill the area, respecting its original aspect ratio. It won’t add extra white space. Here is an example:

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/some_image"
    android:adjustViewBounds="true"/>
```

Regarding iOS, yes, it has the same behavior. When iOS scales the image and its width exceeds the view’s width, it will stretch it to the width of the view, while maintaining the original aspect ratio. It won’t add any white space above or below the image. And there is no need to set the layout size to MATCH_PARENT, as by default it will take all available space. However, it is possible that you still see white space if you are manually setting any other constraints or attributes that affect the image’s dimensions. In iOS, you can also use the Auto Layout feature to set constraints for your ‘UIImageView‘ to avoid any white spaces. The key points are:
- Set ‘UIImageView’s contentMode‘ to ‘.scaleAspectFit‘. It adjusts the view to the given layout bounds while keeping the aspect ratio. - Set ‘ImageView’s constraints‘ to fill the parent view in both width and height. - Set the necessary auto layout constraints to position the ‘UIImageView‘.
In both environments, you don’t have to worry about white spaces if you properly set up the constraints and attributes related to the image scaling and positioning. Hope this helps! If you need more details or examples, let me know. I’m happy to provide further assistance.

Qwen2-72b Output

You could set android:scaleType="centerCrop" on your ImageView. This will ensure that your image will be shown in the largest size possible in your space and maintain aspect ratio.
Your options are:
“‘ ImageView.ScaleType.CENTER: Scale the image to fit inside the ImageView’s bounds, centering it. Aspect ratio is maintained.
ImageView.ScaleType.CENTER_CROP: Scale the image to fit inside the ImageView’s bounds, centering it and cropping to fit. Useful for images that are not the same aspect ratio as the ImageView.
ImageView.ScaleType.CENTER_INSIDE: Scale the image so that it fits entirely inside the ImageView. Aspect ratio is maintained, and the image will not be scaled down if its smaller than the ImageView.
ImageView.ScaleType.FIT_CENTER: Scale the image to fit inside the ImageView, centering and using the image’s top and left as a center point. This scale type is like CENTER but ensures that the entire image will be visible.
ImageView.ScaleType.FIT_END: Scale the image to fit inside the ImageView, aligning the image’s right and bottom to the right and bottom of the ImageView. This scale type is like FIT_START but the alignment is at the end of the ImageView.
ImageView.ScaleType.FIT_START: Scale the image to fit inside the ImageView, aligning the image’s top and left to the top and left of the ImageView.
ImageView.ScaleType.FIT_XY: Scale the image to fit the ImageView’s bounds. The image may be stretched to fit in this space.
ImageView.ScaleType.MATRIX: Uses a supplied Matrix object to scale and position the image. The image will be clipped by the ImageView if it extends outside the ImageView bounds.
“‘
See [the documentation](http://developer.android.com/reference/android/widget/ImageView.ScaleType.html) for a description of each one.