Adam's Law: Textual Frequency Law on Large Language Models

Hongyuan Adam Lu[♣*], **Z.L.**[♣*], **Zefan Zhang**[♣],
Bowen Cao[♡], **Wai Lam**[♡]
♣FaceMind Corporation
♡The Chinese University of Hong Kong
hongyuanlu@outlook.com

Abstract

While textual frequency has been validated as relevant to human cognition in reading speed, its relatedness to Large Language Models (LLMs) is seldom studied. We propose a novel research direction in terms of textual data frequency, which is an understudied topic, to the best of our knowledge. Our framework is composed of three units. First, this paper proposes **T**extual **F**requency **L**aw (TFL), which indicates that frequent textual data should be preferred for LLMs for both prompting and fine-tuning. Since many LLMs are closed-source in their training data, we propose using online resources to estimate the sentence-level frequency. We then utilize an input paraphraser to paraphrase the input into a more frequent textual expression. Next, we propose **T**extual **F**requency **D**istillation (TFD) by querying LLMs to conduct story completion by further extending the sentences in the datasets, and the resulting corpora are used to adjust the initial estimation. Finally, we propose **C**urriculum **T**extual **F**requency **T**raining (CTFT) that fine-tunes LLMs in an increasing order of sentence-level frequency. Experiments are conducted on our curated dataset **T**extual **F**requency **P**aired **D**ataset (TFPD) on math reasoning, machine translation, commonsense reasoning and agentic tool calling. Results show the effectiveness of our framework.[1]

1 Introduction

Large language models (LLMs) have demonstrated many exciting abilities and applications, such as chain-of-thought reasoning (Wang et al., 2023; Wei et al., 2024), machine translation (Lu et al., 2023; Zhu et al., 2024a), and spatial reasoning (Hu et al., 2024), etc. More recently, increasing the length of the reasoning processes has become another popular research direction (DeepSeek-AI et al., 2025; Muennighoff et al., 2025). Another important factor for training is the order of training, where it could be preferable from easy to hard in terms of the data difficulty (Lu and Lam, 2023), or from short to long in terms of data length (Zhu et al., 2025). Yet, what kind of data should be favourable during the training is an overlooked topic. Previous works have explored and concluded that the qual-

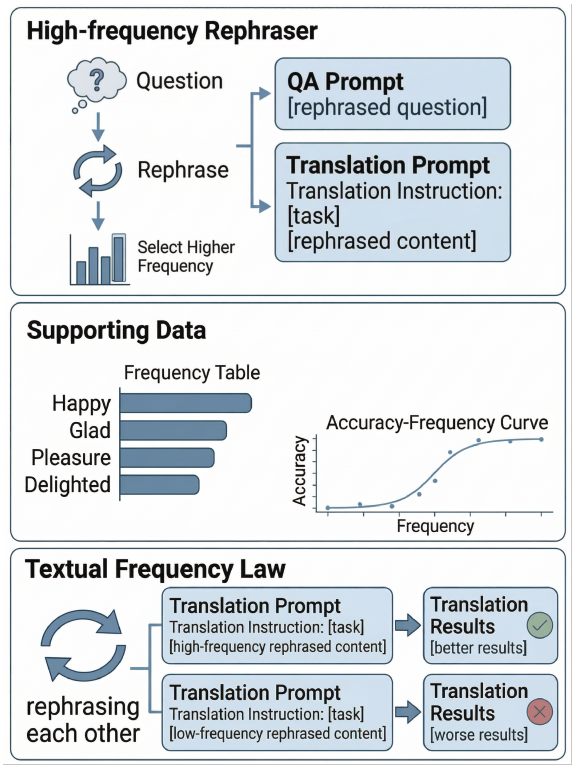

Figure 1: **Top:** A simplified example of use case of Textual Frequency Law, where the prompt contents are rephrased and the prompt contents with higher frequency are selected. **Middle:** We achieve this by estimating sentence-level frequency with word-level frequency. **Bottom:** A toy example showing the effectiveness of our framework. Real case studies are available in the Appendix in Figure 6. The paraphrasing can lead to semantic drift, which is the reason why human annotation is necessary in this process.

[*] Equal Contribution.
[1] https://github.com/HongyuanLuke/frequencylaw

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4079–4105

ity of the data is usually important (Iskander et al., 2024; Jin and Wang, 2024). The amount of data is also important (Grattafiori et al., 2024).

Oh et al. (2024) found that larger models predict rare words better. In the era of LLMs, scaling factors usually mean that larger models can be better. This then may mean that predicting rare (less frequent) words could be a harder task than predicting frequent words. Cao et al. (2024) demonstrated that when prompting LLMs, different prompts with the same meaning could give very different results in terms of quality.

This motivates us to investigate when the data are paraphrased to each other with the same meaning but different language expressions. The use of paraphrases has been explored in NLP research for many cases, such as mitigating data contamination (Zhu et al., 2024b), evaluating generation tasks (Tang et al., 2024) and data augmentation (DA, Abaskohi et al. (2023)). As a DA method, paraphrases are useful for training LLMs (Lu and Lam, 2023), so this means that we might want to include all the paraphrases in the training when it is affordable. However, training resources are usually limited, and we investigate whether the frequency matters when the meaning is kept, and the computational resources are limited for fine-tuning. Also, such investigation on paraphrased inputs into LLMs can be important, as Cao et al. (2024) has found that they usually give different performance, but there isn't a clear conclusion yet which factors are relevant to this phenomenon.

In contrast, this paper proposes novel **T**extual **F**requency **L**aw (TFL), which suggests that when the meanings are kept the same, data with higher sentence-level frequency should be preferred to the ones with low frequency, for both prompting and fine-tuning. The underlying motivation is that this paper postulates that higher-frequency data occurs more frequently than lower-frequency data in the pre-training stage, so they are easier to understand by LLMs. Based on such a law, this paper proposes to calculate the frequency estimation through online open-source data corpora, as many LLMs are closed-source and we usually do not have direct access to their training data. To further enhance the estimation, this paper proposes a novel method called **F**requency **T**extual **D**istillation. TFD conducts story completion with a text dataset on the target LLMs, and the completed story generation is used to enhance the original frequency estimation. Last, we propose **C**urriculum **T**extual **F**requency **T**raining (CTFT) that fine-tunes LLMs in increasing order of sentence-level frequency with the training data, which yields better results.

Our frequency training framework is composed of three units, and our contributions are three-fold:

- We propose **T**extual **F**requency **L**aw, which suggests that high-frequency textual data should be preferred for LLMs when conducting prompting and fine-tuning, when the meaning of the data is kept the same, i.e., they are paraphrases.
- We propose a novel method called **T**extual **F**requency **D**istillation to further enhance the frequency estimation (collected from online resources) via conducting story completion to collect model generation from those LLMs that we do not have direct access to the training textual data.
- We propose a novel method called **C**urriculum **T**extual **F**requency **T**raining that fine-tunes LLMs in an increasing order of sentence-level frequency with the training data.

Figure 1 demonstrates a use case of our proposed framework, where prompts are rephrased to achieve higher accuracy.

2 Prior Works

2.1 Textual Frequency

Textual frequency is even related to human neural activation. Desai et al. (2020) explored the neural activation differences between low-frequency words and high-frequency words in reading tasks, finding that high-frequency words generally evoke stronger neural responses. Alexandrov et al. (2011) explored the neural activation differences between low-frequency words and high-frequency words in reading tasks, finding that high-frequency words generally evoke stronger neural responses. Mohan and Weber (2019) also mentioned the impact of word frequency on semantic retrieval.

Then, textual frequency plays an important role in artificial intelligence. Heylen et al. (2008) investigated the semantic similarity between words of different frequencies and found that high-frequency target words have higher semantic similarity with their nearest neighbour words. Oh et al. (2024) found that larger models predict rare words better. This then may mean that predicting rare (less frequent) words could be a harder task than predicting

frequent words, as larger models can usually be stronger. More recnetly, Lu et al. (2025) discovered that low-frequency words should be explained to LLMs for better machine translation.

2.2 Paraphrasing on Language Models

Paraphrasing is an important language task that is tackled well by language models (Witteveen and Andrews, 2019; Goyal and Durrett, 2020). Yet, paraphrasing can still be a useful method to improve language models from various aspects. Tang et al. (2024) uses paraphrases to generate diverse references, which helps in evaluating language models. Zhu et al. (2024b) uses paraphrasing as a method to cleanly evaluate the possibly contaminated large language models. Gao et al. (2020) uses paraphrases as data augmentation to improve goal-oriented dialogue systems. More recently, Guo et al. (2023) also uses generative data augmentation, which reflects the usefulness of paraphrasing in enhancing model performance. One setting in this paper compares the performance of LLMs on paraphrases with the same meaning but different frequencies. Yet, there are some overlooks in the previous setting. It is crucial as the computational budgets for training and prompting (Cao et al., 2024) are usually limited. It raises questions: which paraphrases are more useful? Should we use all paraphrases?

3 Proposed Approach

3.1 Task Formulation

The large language model (LLM) can be regarded as a Seq2Seq neural network (Sutskever et al., 2014) to follow the instructions to conduct various tasks with additional inputs by maximising the following likelihood:

$$P\left(\mathbf{y} \mid \mathbf{i}, \mathbf{x}\right) = \prod_{j=1}^{\mathbb{T}} P\left(y_j \mid y_1, ..., y_{j-1}, \mathbf{i}, \mathbf{x}\right), \quad (1)$$

where $\mathbb{T}$ represents the length of the generated output and y_j represents the word at the position j that has been inferenced. $\mathbf{i}$ represents the instruction to guide the LLMs to process the inputs. $\mathbf{x}$ represents the source sentences. Note that the actual format could be case by case for different tasks. For example, we conduct experiments on math reasoning and machine translation (MT). For math reasoning, there is no $\mathbf{x}$, as the instruction itself already contains the actual question. In contrast, for MT, $\mathbf{i}$ is usually the instruction to ask LLMs to translate the actual sentence $\mathbf{x}$ to the target language while maintaining the actual meaning. For convenience, we denote $\mathbf{x}$ as the concatenation of the instruction and the actual input in the rest of this section.

3.2 Textual Frequency Law

This paper proposes **T**extual **F**requency **L**aw (TFL) to select the paraphrases with the highest sentence-level textual frequency for both prompting and fine-tuning on LLMs:

$$\text{argmax}_{\mathbf{x} \in \mathcal{P}}(\text{sfreq}(\mathbf{x}, \mathcal{D})), \quad (2)$$

where $\mathbf{x}$ corresponds to the textual input as in Equation 1. $\mathcal{P}$ represents a set of paraphrases that contain the same meaning. sfreq represents a function that evaluates a sentence-level textual frequency.

Such a frequency function freq can be obtained and calculated given a corpus $\mathcal{D}$. In this paper, we suggest that such sentence-level frequency can be estimated by using a position-unaware multiplication of word-level frequency:

$$\text{sfreq}(\mathbf{x}, \mathcal{D}) = \sqrt[\mathbb{K}]{\frac{1}{\prod_{k=1}^{\mathbb{K}} \text{wfreq}(\mathbf{x}_k, \mathcal{D})}} \quad (3)$$

Here, wfreq is the word-level frequency calculator that we use to estimate the sentence-level frequency. In this paper, we suggest that there is no need to obtain the actual training data of LLMs, and an arbitrary text corpus can be adapted to obtain the frequency. We obtain the sentence-level frequency with the inverse normalised multiplication of the word-level frequency.

Prompting When prompting with $\mathbf{x}$, higher $\mathbf{x}$ should be used to generate outputs from LLMs.

Fine-tuning For fine-tuning, $\mathbf{x}$ with a higher frequency should be used together with the desired ground truth output $\mathbf{y}$ to fine-tune the LLMs.

3.3 Textual Frequency Distillation

Note that the frequency we obtained in the previous section is an estimation from online resources but not the actual data, as many LLMs are closed-source in their training data. This paper proposes **T**extual **F**requency **D**istillation (TFD) to further enhance this estimation. TFD asks LLMs to generate data by the following instructions:

> Please conduct story completion on the following data: <textual data>

, where <textual data> represents the data we have in our training set. We denote this distilled dataset as $\mathcal{D}'$. We obtain a new frequency estimation:

$$\mathcal{F}_2 = \text{sfreq}(\mathbf{x}, \mathcal{D}'), \tag{4}$$

and we denote the original frequency estimation as in Equation 3 as $\mathcal{F}_1$. Note that this step in obtaining $\mathcal{F}_2$ is relatively computationally expensive, as the data are distilled from the actual LLMs. This is, therefore, optional, and our proposed method is still effective even with $\mathcal{F}_1$ only. We then calculate the final frequency $\mathcal{F}$ as:

$$\mathcal{F}(x) = \alpha\mathcal{F}_1(x) + (1 + \zeta\mathbb{1}(\mathcal{F}_1(x) = 0))\beta\mathcal{F}_2(x), \tag{5}$$

where α, β, and ζ are hyper-parameters. In the formula above, ζ is a strengthening factor to increase the effect of the distilled frequency when the words yield an ignorable frequency in the original estimation from $\mathcal{F}_1$. The calculated frequency $\mathcal{F}(x)$ is then used to choose the highest frequency instead of the original estimated frequency as in Equation 2 and Equation 3.

3.4 Curriculum Textual Frequency Training

Motivated by the fact that low-frequency expressions can be more diverse (Lu and Lam, 2023), which should be trained first (Jiang et al., 2014), we propose **C**urriculum **T**extual **F**requency **T**raining (CTFT), a method that further uses the frequency information beyond paraphrase selection during prompting. For a training set $\mathcal{T}$ that is composed of $\mathbb{N}$ instances, we propose to arrange the data in the following training order for each epoch:

$$\text{sort}_{x_n \in \mathcal{T}}(\mathcal{F}(x_n)), \tag{6}$$

where sort is a sorting function that arranges the order from lower frequency sentence-level to higher sentence-level frequency for each training instance x_n in $\mathcal{T}$ with a total number of $\mathbb{N}$ instances. Note that the training instances are usual machine learning datasets here and do not have to be paraphrases of each other. We experiment with CTFT on the fine-tuning scenarios on LLMs. CTFT extends TFL and TFD to a better fine-tuning scenario.

3.5 Textual Frequency Paired Dataset

There is almost no such dataset for our the goal. Therefore, we collect our own dataset, **T**extual **F**requency **P**aired **D**ataset (TFPD), for this paper. Based on the original datasets GSM8K (Cobbe et al., 2021), FLORES-200 (NLLB-Team, 2022), CommonsenseQA (Talmor et al., 2019), and ToolBench (Guo et al., 2024), we use GPT-4o-mini to rephrase the English sentences in GSM8K and FLORES-200. The rephrased sentences are sent to three human annotators. For human annotation, we hired three experienced annotators who have degrees relevant to English Linguistics, paid with reasonable payment, to conduct a human validation on the generated sentences. We discard the instances if the three sentences do not have the same meaning by any human annotator. We use the following instructions to rephrase the datasets automatically:

> My goal is to transform the original sentence into both more common and less common expressions.
> Note: Do not omit any words such as verbs, adjectives, nouns, or adverbs.
> You must generate two types of sentences:
> (1) ten sentences using less common, more complex words.
> (2) ten sentences using more common, simpler words.
> Return all 20 sentences directly, separated by |||| and do not use numbering.
> Original sentence: sentence

Tasks	MR	MT	CR	TC
high-frequency				
#. Sentences	738	526	575	114
Avg Length	25.86	21.70	23.66	41.96
Max Length	71	60	64	73
Min Length	11	7	9	22
low-frequency				
#. Sentences	738	526	575	114
Avg Length	25.28	24.78	22.43	47.82
Max Length	59	62	57	86
Min Length	10	9	8	25

Table 1: Statistics of **T**extual **F**requency **P**aired **D**ataset (TFPD). We denote Math Reasoning as **MR**, Machine Translation as **MT**, Commonsense Reasoning as **CR**, and Tool Calling as **TC**. We denote the total instances in the dataset as #. Sentences, and we report the length in English words. The ground-truth answer from the original datasets is directly adopted without modification. Each sentence in the high-frequency partition is paired with one sentence in the low-frequency partition.

The above instructions on GPT-4o-mini then generate 20 paraphrases. We select the two sentences with the lowest and highest frequency, respectively, as in Equation 1. Those two sentences are sent along with the original input sentence for succeeding human annotation to check whether all three sentences have the same meaning:

- The same meaning: I believe these three sentences have the same meaning.
- Maybe the same meaning: Maybe these three sentences have the same meaning, but I might be wrong because of some reasons, for example, some rephrased words might not be appropriate for the context.
- Not the same meaning: I am sure that these three sentences do not have the same meaning.

We only preserve those samples that all our annotators believe are authentically the same meaning. Finally, we obtain 738 pairs out of 1,319 original GSM8K test instances, and we obtain 526 pairs out of 1,012 original FLORES-200 dev-test instances. Note that for the fine-tuning experiments, we use the constructed TFPD dataset as the training data to check the impact of textual frequency on fine-tuning, and we randomly select 500 samples from the FLORES-200 dev set for evaluation. This process is approved by FaceMind ethics review aboard.

Table 1 presents the length statistics of the samples. For space reasons, we present frequency statistics in Appendix in Table 17.

4 Experimental Setup

4.1 Evaluation Metrics

For the task of math reasoning, accuracy is adopted as the evaluation metric (Cobbe et al., 2021). For the task of machine translation, we report the chrF (Popović, 2015) and the BLEU (Papineni et al., 2002) evaluations provided by the sacreBLEU repository.[2] We also adopt neural-based evaluation using COMET scores versioned wmt22-comet-da[3] (Rei et al., 2020). Note that there are 37 supported languages by COMET, out of 100 languages in this study. We release the full list as in Appendix. We use chrF signature of the parameters with nworde=6, ncorder=6, beta=2. We use BLEU signature of ngram=4, weights=(0.25, 0.25, 0.25, 0.25), smoothing=method1, smoothingfunction=SmoothingFunction().method1, tokenizer=nltkwordtokenize.

4.2 Baselines

We conduct experiments on both closed-source and open-source LLMs for better reproducibility on GPT-4o-mini and DeepSeek-V3 (DeepSeek-AI et al., 2024). DeepSeek-V3 is an MoE model with 671B model parameters. Both of them are widely used LLMs with robust multilingual translation capabilities. We also use doubao-1.5-pro-32k and qwen2.5-7b-instruct as baselines for our translation experiments. For the fine-tuning experiments validating the effectiveness of high-frequency data and the usefulness of CTFT, all experiments are conducted on qwen2.5-7b-instruct, which is an open-source LLM. We use Llama-3.3-70B-Instruct in our MR experiments (Grattafiori et al., 2024). We use LoRA fine-tuning (Hu et al., 2022) throughout the paper. The hyperparameters for fine-tuning are presented in Appendix for better reproducibility.

We also compare our method for the reverse setting (fine-tuning from high-frequenty to low-frequency) as well as traditional curriculum learning (from easy-to-hard, (Lu and Lam, 2023)). For the easy-to-hard baseline, we use Max Dependency Tree Depth as the difficulty function.[4]

4.3 Off-the-shelf Frequency Estimation

For off-the-shelf frequency estimation, we adopt off-the-shelf resources for estimation [5] using Zipf frequency (Speer, 2022). Since this project is further built on many resources such as ParaCrawl (Bañón et al., 2020), we refer the readers to their projects for more references.

4.4 Language Selection

We randomly select 100 languages from the FLORES-200 datasets for our prompting experiments, and we release their language class according to Joshi et al. (2020) in Table 20 in Appendix. More than half of the languages are relatively low-resource according to the class definition (class 0 or class 1). For the experiments on CTFT, we use Kabuverdianu (kea_Latn), Kikuyu (kik_Latn), Pangasinan (pag_Latn), and Standard Latvian (lvs_Latn).

[2] https://github.com/mjpost/sacrebleu

[3] https://github.com/Unbabel/COMET

[4] We use nlp = spacy.load("en_core_web_sm") to calculate it.

[5] https://github.com/rspeer/wordfreq

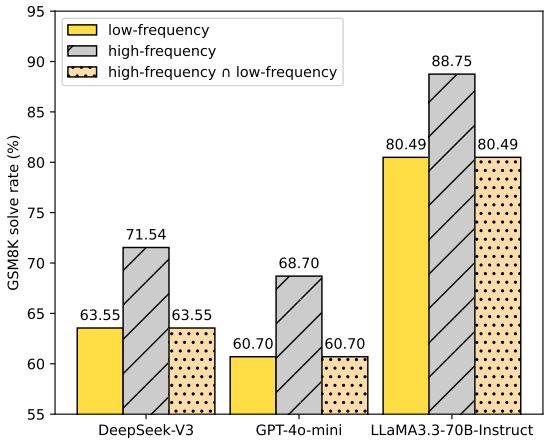

Figure 2: The overall accuracy of TFPD on math reasoning for our proposed framework. It is obvious that the high-frequency partition in TFPD has a higher accuracy than the low-frequency partition. High-frequency ∩ low-frequency denotes a model that is correct in both low-frequency and high-frequency partitions.

4.5 Translation Prompt

We release our 1-shot prompt for translation for better reproducibility:

> Translate the following sentence from English to {lang}.
> For example:
> sentence: Television reports show white smoke coming from the plant.
> translation: {trans}
> Now, please translate the following sentence to {lang}.
> sentence: {question}
> Your output format must be like this:
> The translation result is:

5 Results

5.1 Prompting on Math Reasoning

Figure 2 presents the overall accuracy of TFPD on the task of math reasoning with prompting experiments. Our proposed framework is effective on all models that we experimented on. On DeepSeek-V3, the accuracy goes from 63.55% to 71.54%. On GPT-4o-mini, the accuracy goes from 60.70% to 68.70%. On LlaMA3.3-70B-Instruct, it goes from 80.49% to 88.75%. We also conduct deeper analyses. Specifically, we calculate the intersection of low-frequency and high-frequency partitions. We found that when a sample pair has a correct model generation on its low-frequency partition, its high-frequency version is still correct. In other words, using our proposed framework only improved those samples which were originally answered incorrectly by the models on the low-frequency partition. For those ones which were originally answered correctly by the models on the low frequency partition, their performance is maintained with the high frequency partition.

For space reasons, Table 18 in the Appendix represents that our method is consistently useful and high-frequency data brings improvements on different sizes of qwen-2.5 models across 0.5b to 72b on the task of MR.

Table 21 indicates that the chain-of-thought process is improved, which can be the reason why the math reasoning capabilities are improved.

5.2 Prompting on Neural Machine Translation

Figure 3 demonstrates the results on Neural Machine Translation (NMT) on our TFPD dataset. The orange line indicates the model using the high-frequency partition in our TFPD dataset on ChatGPT or DeepSeek models. The results follow our proposed TFL, which suggests that high-frequency rephrases should be preferred as inputs into LLMs. Specifically, for all six results on all metrics we report and all baselines we conduct, high-frequency partition gives the best results in overall. We also found that ChatGPT and DeepSeek models are close in their translation results on the language pairs we conducted experiments on, as their Figure seems to be relatively similar to each other. This is reasonable, as both of them are strong LLMs. We also report results on 37 languages supported by the COMET model in use. The results also suggest the effectiveness of our proposed law.

Table 3 summarises the improvements on NMT. We can see that when compared to our best baseline using the low-frequency partition, translation on most of the language pairs is improved. For example, 99 out of 100 language pairs are improved for BLEU on DeepSeek-V3. 63 of them are improved by more than 1 point. 31 of them are improved by more than 3 points, and 12 of them are improved by more than 5 points. The observations are consistent across all metrics, namely, BLEU, chrF, and COMET scores we use, across both DeepSeek-V3 and GPT-4o-mini, which suggests the effectiveness of our proposed law. When there is any performance degradation, they are all less than 1 point across the metrics and the models, which enhances

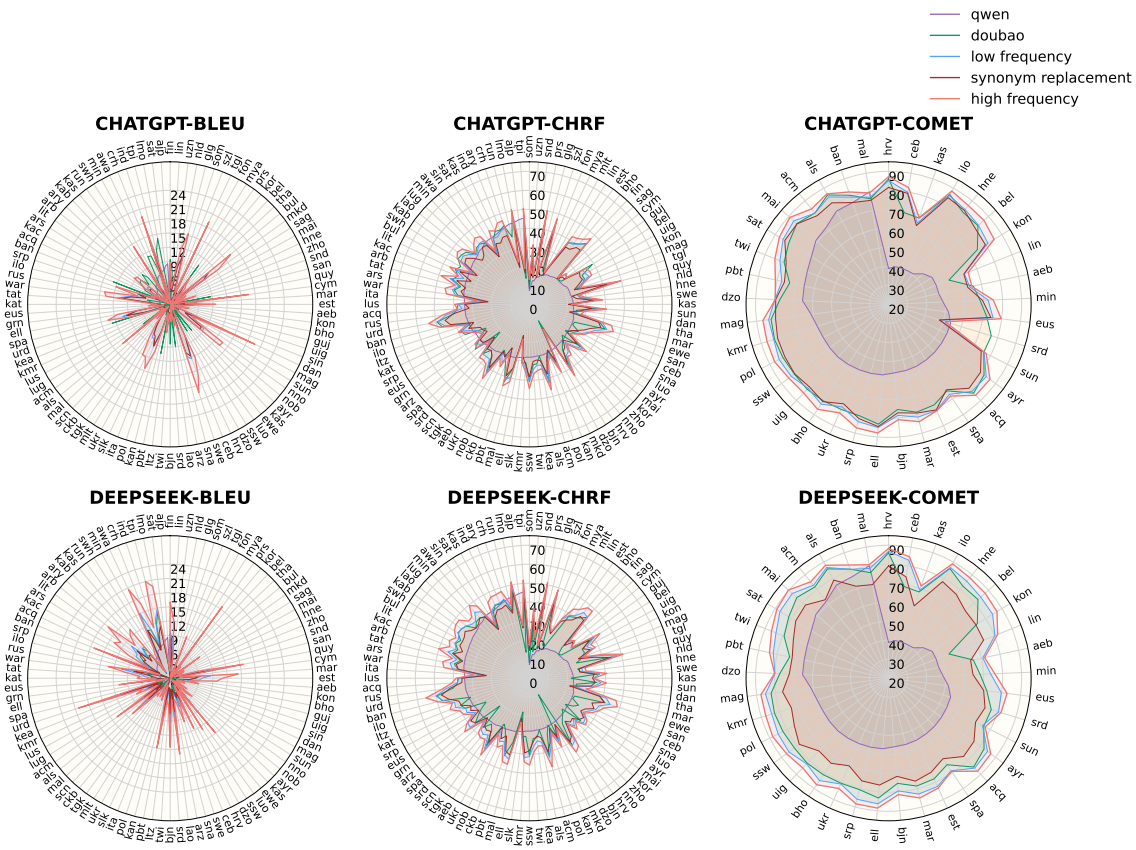

Figure 3: The figure demonstrating the performance of our proposed framework in using high-frequency partition for translation. Results are reports on translating from English into other languages. Detailed numbers are reported in Appendix in Table 8, 9, 10, 11, 12, and 13. Synonym is a baseline that replaces words randomly with their higher-frequency rephrases using NLTK: https://www.nltk.org/.

Models	GPT-4o-mini	DeepSeek-V3	Llama-3.3-70B-Instruct
Low-frequency partition	0.6747	0.7043	0.7530
High-frequency partition	**0.6974**	**0.7235**	**0.7704**

Table 2: Results reported in accuracy on the partition of CR. We see that the high-frequency partition gives better results on all baseline models.

Models	# improved	> 1 pt	> 3 pts	> 5 pts	# degraded	> 1 pt	> 3 pts	> 5 pts
				BLEU				
DeepSeek-V3	99/100	63/99	31/99	12/99	1/100	0/1	0/1	0/1
GPT-4o-mini	95/100	49/95	27/95	5/95	5/100	0/5	0/5	0/5
				chrF				
DeepSeek-V3	100/100	86/100	40/100	7/100	0/100	0/0	0/0	0/0
GPT-4o-mini	91/100	75/91	34/91	2/91	9/100	0/9	0/9	0/9
				COMET				
DeepSeek-V3	37/37	33/37	4/37	0/37	0/37	0/0	0/0	0/0
GPT-4o-mini	36/37	35/36	11/36	0/36	1/37	0/1	0/1	0/1

Table 3: Statistics of the changes on prompting experiments in BLEU, chrF, and COMET scores with the high-frequency partition compared to the low-frequency partition on our established TFDP dataset. We evaluate translation from English into other languages. Most translations have been clearly improved. When there is any degradation, the degradation is less than 1 point. We denote 'point' as 'pt' and 'points' as 'pts'.

Models	kea_Latn	kik_Latn	pag_Latn	lvs_Latn
		BLEU		
Original Model	0.9346	1.0342	1.2296	2.2646
Fine-tuned Model	4.6772	1.2811	4.5129	4.1954
Easy-to-hard Baseline	5.1674	1.3185	4.4955	3.5366
High-to-low Baseline	5.1179	1.5298	4.5365	3.7840
FT on LF w/o CTFT	4.3899	1.4223	3.9073	3.2221
FT on 1/2 LF 1/2 HF w/o CTFT	4.7928	1.4783	4.4291	3.4787
FT on HF w/o CTFT	5.2466	1.2432	3.7781	3.9156
FT on HF w/ CTFT	**5.3992**	**1.6570**	**4.9102**	**4.6027**
		chrF		
Original Model	26.9844	20.6636	29.4351	33.2322
Fine-tuned Model	39.3714	25.6175	34.4672	34.0584
FT on LF w/o CTFT	39.4022	26.2465	33.9848	33.5538
Easy-to-hard Baseline	40.6414	26.4981	35.5396	35.3337
High-to-low Baseline	41.0234	26.5316	35.8125	36.1577
FT on 1/2 LF 1/2 HF w/o CTFT	40.7831	26.8192	35.3375	34.2120
FT on HF w/o CTFT	40.6515	26.4975	33.4990	35.0732
FT on HF w/ CTFT	**41.6206**	**27.7719**	**36.5285**	**37.0171**

Table 4: Results of fine-tuning experiments on translation from English into other languages, tested on the original FLORES-200 benchmark. Fine-tuned Model is tuned on the original FLORES-200 dataset. FT denotes fine-tuning, LF denotes low-frequency, HF denotes high-frequency, and CTFT denotes Curriculum Textual Frequency Training. 1/2 LF 1/2 HF denotes a training set with half samples sampled from the low-frequency partition and half samples sampled from the high-frequency partition. COMET is not reported due to unsupported languages.

our claim and the usefulness of our law.

5.3 Prompting on Commonsense Reasoning

Table 2 reports addtional results on the commonse reasoning partition CR. It clearly shows that the high-frequency part surpasses the low-frequency part. This validates the effectiveness of our method.

5.4 Fine-tuning on Neural Machine Translation

Table 4 presents our results for fine-tuning on NMT. There are three takeaways from this Table.

High-frequency partition is even better than the ground-truth data For the baseline of *Fine-tuned Model*, *FT on HF w/o TFD w/o CTFT* is even better across the languages and the metrics. The former one uses the original FLORES-200 dataset for fine-tuning, and the latter uses our TFPD dataset for fine-tuning without any TFD or CTFT. The improvements are obvious, for example, it improves from 4.6772 (+0%) in BLEU to 5.2466 (+12.17%) in BLEU on kea_Latn.

High-frequency partition is better than the low-frequency partition By looking at the baselines *FT on LF w/o CTFT* and *FT on HF w/o CTFT*. It is first clear that the latter one, using the high-frequency partition, is better than the former one, using a low-frequency partition. Interestingly, replacing half of the low-frequency partition randomly using the high-frequency partition can still obviously improve the results. Specifically, the improvement can be from 3.9073 (+0%) to 4.4291 (+13.35%) in BLEU on pag_Latn.

CTFT is useful for fine-tuning on translation By looking at the baseline *FT on HF w/o CTFT* and *FT on HF w/ CTFT*, the latter one trains the model using CTFT, from the order of low-to-high in terms of the textual frequency. This yields 8/8 of the best metrics we got in all the experiments. Specifically, the improvement can be from 3.7781 (+0%) to 4.9102 (+29.96%) in BLEU on pag_Latn.

5.5 Analysis on TFD

Figure 4 presents the ablation study on TFD. It is obvious that removing TFD causes a drop in performance. For example, 100% of the language pairs are better with TFD on COMET scores with DeepSeek-V3. This validates the usefulness of TFD. Figure 5 also demonstrates the relationship

Metric	High-Frequency	Low-Frequency	Δ(HF-LF)	Pearson Corr.	Spearman Corr.
Math Reasoning					
Max Dependency Tree Depth	5.02	5.72	-0.70	-0.0447	-0.0285
Mean Dependency Distance	2.12	2.22	-0.10	-0.0086	0.0094
Flesch-Kincaid Grade Level	4.36	6.35	-1.99	-0.0799	-0.0545
Machine Translation					
Max Dependency Tree Depth	5.52	7.51	-1.99	-0.2713	-0.2822
Mean Dependency Distance	2.31	2.47	-0.16	-0.1137	-0.1257
Flesch-Kincaid Grade Level	8.97	9.08	-0.11	-0.1673	-0.1528

Table 5: Textual complexity metrics and their correlation with frequency. Corr. denotes correlation. We use nlp = spacy.load("en_core_web_sm") for calculation.

Bin Range	N	BLEU(HF)	BLEU(LF)	ΔBLEU(HF-LF)	chrF(HF)	chrF(LF)	ΔchrF(HF-LF)
Strict Depth Match	144	20.82	16.04	+4.78	48.73	43.86	+4.87
[0%, 5%)	144	20.82	16.04	+4.78	48.73	43.86	+4.87
[5%, 10%)	6	22.45	14.79	+7.65	49.76	49.19	+0.57
[10%, 15%)	71	19.12	15.38	+3.74	46.19	44.71	+1.47
[15%, 20%)	65	20.93	14.77	+6.16	48.91	43.46	+5.45
[20%, 25%)	53	24.08	18.52	+5.56	50.87	44.27	+6.60
[25%, 30%)	65	19.75	12.54	+7.21	47.53	42.51	+5.01
[30%, 35%)	41	19.90	12.61	+7.29	47.78	43.72	+4.05
[35%, 40%)	17	19.03	14.13	+4.90	44.22	42.62	+1.60
[40%, 45%)	28	16.53	9.76	+6.77	46.47	40.92	+5.55
[50%, 55%)	21	13.89	16.20	-2.31	41.65	46.86	-5.21
[55%, 60%)	9	10.93	3.33	+7.60	45.62	38.92	+6.70
[60%, 65%)	4	17.13	12.18	+4.95	43.30	44.46	-1.16
[65%, 70%)	2	15.54	4.36	+11.17	37.17	39.72	-2.56

Table 6: Separated bins with those high-frequency and low-frequency samples with restricted tree depth difference.

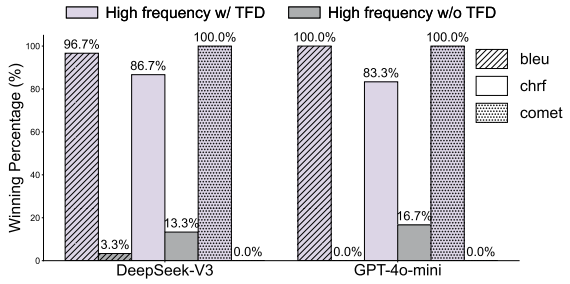

Figure 4: The ablation study results of TFD on TFPD. The results are compared on BLEU, chrf and COMET. The bars are plotted in terms of the winning percentages.

between the amount of data used for frequency distillation and the performance improvement. Overall, with more data used for TFD, there is a greater performance gain. This further validates the usefulness of TFD. Finally, combining prompting with higher-frequency paraphrases on models with CTFT as a whole framework is useful, as presented in the Appendix in Table 15.

5.6 Correlation on Frequency

For space reasons, we present a correlation analysis between textual frequency and final translation performance, even when the instances are not paraphrases to each other using the full translation dataset in TFPD. We present the final results in Appendix in Table 16. There is are strong correlation (1.0) on multiple languages when translating from English. This strengths our claim.

Table 5 represents the relationship between textual complexity and frequency, and we see that they have very weak correlation. This enhances the usefulness of our method by distingushing TFL from the traditional curriculum learning. Table 6 shows that in most bins, high-frequency prompts are better. Only in 1 bin [50%-55%], the low-frequency prompts are better on BLEU and chrF, but there are only 21 samples in this bin. This means that high-frequency prompts are consistently better.

Finally, we present a theoretical proof in Appendix to strength our claim.

6 Conclusions

This paper proposed a framework for textual frequency on LLMs, which is composed of three units, namely TFL, TFD, and CTFT. High-frequency inputs are suggested by our framework, in both tuning and training on LLMs, which can be combined with curriculum learning to improve final performance. We conduct experiments on tasks of Math Reasoning, Machine Translation on hundreds of language pairs, Commonsense Reasoning, and Agentic Tool Calling. Experimental results and extensive analysis suggest the effectiveness of our textual frequency framework. Extensive analysis indicates that when inputs are even different, the final outputs of LLMs are positively related to textual frequency, which further suggests the soundness of our proposed framework.

Limitations

Using story completion to obtain frequency estimation can bring certain computational costs, yet this workaround the necessity of obtaining closed-resourced training corpora of LLMs, which is often unrealistic.

Ethical Statement

We honour and support the ACL ARR Code of Ethics. The datasets used in this work are well-known and widely used, and the dataset preprocessing does not make use of any external textual resource. In our view, there is no known ethical issue. End-to-end pre-trained LLMs are also used, which are subjected to generating offensive context. But the above-mentioned issues are widely known to commonly exist for these models. Any content generated do not reflect the view of the authors.

References

Amirhossein Abaskohi, Sascha Rothe, and Yadollah Yaghoobzadeh. 2023. LM-CPPF: Paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 670–681, Toronto, Canada. Association for Computational Linguistics.

A. Alexandrov, D. Boricheva, F. Pulvermüller, and Y Shtyrov. 2011. Strength of word-specific neural memory traces assessed electrophysiologically. *PLoS ONE*, 6(8):e22999.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Bowen Cao, Deng Cai, Zhisong Zhang, Yuexian Zou, and Wai Lam. 2024. On the worst prompt performance of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *arXiv e-prints*, arXiv:2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv e-prints*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang,

Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. DeepSeek-V3 Technical Report. *arXiv e-prints*, arXiv:2412.19437.

Rutvik H Desai, Wonil Choi, and John M Henderson. 2020. Word frequency effects in naturalistic reading. *Language, cognition and neuroscience*, 35(5):583—594.

David Freedman, Robert Pisani, and Roger Purves. 2007. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 639–649, Online. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252, Online. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, and Tobias Speckbacher. 2024. The Llama 3 Herd of

Models. *arXiv e-prints*, arXiv:2407.21783.

Zhen Guo, Peiqi Wang, Yanwei Wang, and Shangdi Yu. 2023. Dr. llama: Improving small language models in domain-specific qa via generative data augmentation.

Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. 2024. StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11143–11156, Bangkok, Thailand. Association for Computational Linguistics.

Yanjin He, Qingkai Zeng, and Meng Jiang. 2025. Pre-trained models perform the best when token distributions follow Zipf's law. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28009–28021, Suzhou, China. Association for Computational Linguistics.

Kris Heylen, Yves Peirsman, Dirk Geeraerts, and Dirk Speelman. 2008. Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hanxu Hu, Hongyuan Lu, Huajian Zhang, Yun-Ze Song, Wai Lam, and Yue Zhang. 2024. Chain-of-symbol prompting for spatial reasoning in large language models. In *First Conference on Language Modeling*.

Shadi Iskander, Sofia Tolmach, Ori Shapira, Nachshon Cohen, and Zohar Karnin. 2024. Quality matters: Evaluating synthetic data for tool-using LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4958–4976, Miami, Florida, USA. Association for Computational Linguistics.

Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander G. Hauptmann. 2014. Self-paced learning with diversity. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2078–2086, Cambridge, MA, USA. MIT Press.

Jing Jin and Houfeng Wang. 2024. Select high-quality synthetic QA pairs to augment training data in MRC under the reward guidance of generative language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14543–14554, Torino, Italia. ELRA and ICCL.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2023. Transformer language models handle word frequency in prediction head. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4523–4535, Toronto, Canada. Association for Computational Linguistics.

Hongyuan Lu and Wai Lam. 2023. PCC: Paraphrasing with bottom-k sampling and cyclic learning for curriculum data augmentation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 68–82, Dubrovnik, Croatia. Association for Computational Linguistics.

Hongyuan Lu, Zixuan Li, Zefan Zhang, and Wai Lam. 2025. SLoW: Select low-frequency words! automatic dictionary selection for translation on large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 898–913, Suzhou, China. Association for Computational Linguistics.

Hongyuan Lu, Haoran Yang, Haoyang Huang, Dongdong Zhang, Wai Lam, and Furu Wei. 2023. Chain-of-Dictionary Prompting Elicits Translation in Large Language Models. *arXiv e-prints*, arXiv:2305.06575.

Nikolay Mikhaylovskiy. 2025. Zipf's and heaps' laws for tokens and LLM-generated texts. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15469–15481, Suzhou, China. Association for Computational Linguistics.

Ranjini Mohan and Christine Weber. 2019. Neural activity reveals effects of aging on inhibitory processes during word retrieval. *Aging, Neuropsychology, and Cognition*, 26(5):660–687. PMID: 30223706.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv e-prints*, arXiv:2501.19393.

NLLB-Team. 2022. No language left behind: Scaling human-centered machine translation.

Byung-Doh Oh, Shisen Yue, and William Schuler. 2024. Frequency explains the inverse correlation of large language models' size, training data amount, and surprisal's fit to reading times. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2644–2663, St. Julian's, Malta. Association for Computational Linguistics.

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

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

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

Tianyi Tang, Hongyuan Lu, Yuchen Jiang, Haoyang Huang, Dongdong Zhang, Xin Zhao, Tom Kocmi, and Furu Wei. 2024. Not all metrics are guilty: Improving NLG evaluation by diversifying references. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6596–6610, Mexico City, Mexico. Association for Computational Linguistics.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 215–220, Hong Kong. Association for Computational Linguistics.

Wenhao Zhu, Pinzhen Chen, Hanxu Hu, Shujian Huang, Fei Yuan, Jiajun Chen, and Alexandra Birch. 2025. Generalizing From Short to Long: Effective Data Synthesis for Long-Context Instruction Tuning. *arXiv e-prints*, arXiv:2502.15592.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024a. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

Wenhong Zhu, Hongkun Hao, Zhiwei He, Yun-Ze Song, Jiao Yueyang, Yumeng Zhang, Hanxu Hu, Yiran Wei, Rui Wang, and Hongyuan Lu. 2024b. CLEAN–EVAL: Clean evaluation on contaminated large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 835–847, Mexico City, Mexico. Association for Computational Linguistics.

Appendix

Supported Languages by COMET		
ell_Grek	spa_Latn	bel_Cyrl
acm_Arab	hrv_Latn	mar_Deva
srp_Cyrl	uig_Arab	est_Latn
pol_Latn	ukr_Cyrl	eus_Latn
ajp_Arab	mkd_Cyrl	swe_Latn
urd_Arab	ind_Latn	swh_Latn
uzn_Latn	fin_Latn	ita_Latn
kor_Hang	lao_Laoo	rus_Cyrl
arb_Arab	bul_Cyrl	nld_Latn
san_Deva	ars_Arab	lit_Latn
tha_Thai	glg_Latn	slk_Latn
cym_Latn	dan_Latn	snd_Arab
som_Latn	-	-

Table 7: The list of 37 languages supported by our COMET model for evaluation on machine translation.

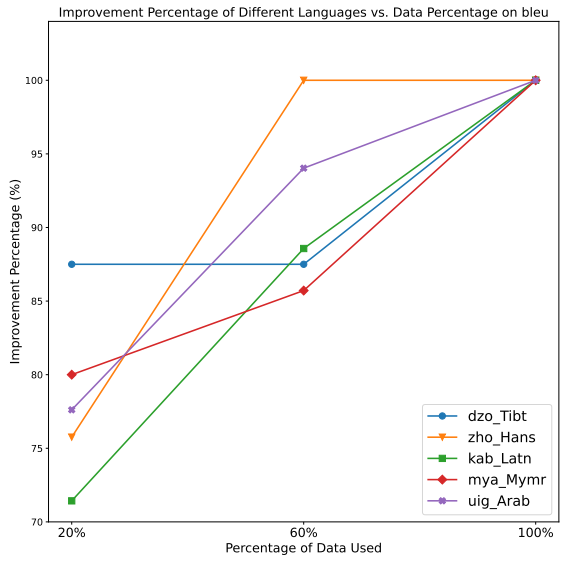

Figure 5: The figure that demonstrates the relationship between performance percentage and the amount of data used for TFD. We can see that with more data used, the performance improvement increases.

Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High
acm_Arab	2.54	**3.29**	acq_Arab	3.07	**4.51**	aeb_Arab	2.36	**3.22**	ajp_Arab	2.7	**4.14**	als_Latn	9.36	**14.54**
arb_Arab	6.15	**9.24**	ars_Arab	5.2	**6.36**	ary_Arab	0.73	**0.98**	arz_Arab	3.17	**4.66**	awa_Deva	1.93	**3.06**
ayr_Latn	0.41	**0.59**	ban_Latn	3.7	**4.24**	bel_Cyrl	3.7	**5.66**	bho_Deva	3.38	**4.43**	bjn_Latn	5.31	**6.08**
bul_Cyrl	10.61	**16.97**	ceb_Latn	13.06	**17.26**	ckb_Arab	1.54	**2.47**	crh_Latn	1.67	**2.32**	cym_Latn	15.06	**20.64**
dan_Latn	12.58	**21.04**	dzo_Tibt	0.03	**0.04**	ell_Grek	7.97	**12.07**	est_Latn	6.1	**9.81**	eus_Latn	3.94	**6.09**
ewe_Latn	1.18	**1.54**	fin_Latn	5.42	**9.44**	fon_Latn	0.26	**0.39**	glg_Latn	11.01	**16.84**	grn_Latn	1.6	**2.0**
guj_Gujr	4.17	**6.97**	hne_Deva	2.08	**2.32**	hrv_Latn	8.74	**13.83**	ilo_Latn	7.99	**10.89**	ind_Latn	13.91	**20.26**
ita_Latn	9.86	**15.95**	kab_Latn	1.03	**1.21**	kac_Latn	1.36	**1.72**	kan_Knda	2.71	**4.57**	kas_Arab	0.4	**0.55**
kas_Deva	0.16	**0.32**	kat_Geor	2.77	**4.72**	kea_Latn	3.84	**5.58**	kmr_Latn	2.45	**3.03**	kon_Latn	3.16	**4.43**
kor_Hang	2.95	**5.02**	lao_Laoo	1.46	**1.73**	lin_Latn	3.83	**5.06**	lit_Latn	6.62	**10.15**	lmo_Latn	2.99	**3.77**
ltz_Latn	6.69	**10.03**	lug_Latn	1.47	**2.04**	luo_Latn	1.04	**1.4**	lus_Latn	2.78	**3.26**	mag_Deva	4.2	**5.24**
mai_Deva	2.97	**3.97**	mal_Mlym	2.12	**3.22**	mar_Deva	3.04	**5.03**	min_Latn	6.05	**7.8**	mkd_Cyrl	10.06	**14.87**
mlt_Latn	7.73	**11.17**	mya_Mymr	0.54	**0.66**	nld_Latn	8.65	**12.27**	nno_Latn	8.55	**14.01**	nob_Latn	8.54	**12.84**
pbt_Arab	2.29	**3.08**	pol_Latn	6.74	**11.03**	prs_Arab	5.27	**7.65**	quy_Latn	0.55	**0.68**	run_Latn	1.47	**2.05**
rus_Cyrl	9.42	**16.08**	sag_Latn	0.67	**0.9**	san_Deva	0.11	**0.38**	sat_Olck	1.0	**1.71**	scn_Latn	4.08	**5.94**
sin_Sinh	2.46	**3.96**	slk_Latn	8.49	**13.31**	sna_Latn	2.94	**4.54**	snd_Arab	4.35	**6.15**	som_Latn	2.5	**3.25**
spa_Latn	10.36	**14.89**	srd_Latn	7.44	**11.66**	srp_Cyrl	8.61	**14.29**	ssw_Latn	1.25	**1.53**	sun_Latn	5.6	**7.42**
swe_Latn	11.7	**18.63**	swh_Latn	10.76	**15.63**	szl_Latn	3.45	**5.02**	tat_Cyrl	3.75	**5.86**	tgk_Cyrl	4.5	**6.25**
tgl_Latn	14.73	**19.68**	tha_Thai	0.95	**1.3**	tpi_Latn	8.8	**10.85**	twi_Latn	2.44	**3.08**	uig_Arab	1.5	**2.27**
ukr_Cyrl	7.8	**12.75**	urd_Arab	6.26	**9.62**	uzn_Latn	4.1	**5.98**	war_Latn	10.42	**13.29**	zho_Hans	**0.42**	0.26

Table 8: Results on DEEPSEEK-V3 in BLEU scores on 100 languages from English into other languages.

Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High
acm_Arab	36.94	**38.42**	acq_Arab	37.07	**39.17**	aeb_Arab	34.44	**36.79**	ajp_Arab	37.79	**40.58**	als_Latn	42.09	**45.98**
arb_Arab	40.87	**45.4**	ars_Arab	39.43	**41.77**	ary_Arab	30.51	**31.48**	arz_Arab	36.87	**39.65**	awa_Deva	30.45	**31.84**
ayr_Latn	28.95	**30.13**	ban_Latn	36.79	**38.27**	bel_Cyrl	35.88	**38.77**	bho_Deva	31.92	**33.93**	bjn_Latn	41.31	**42.63**
bul_Cyrl	45.11	**50.17**	ceb_Latn	46.96	**49.72**	ckb_Arab	38.68	**40.8**	crh_Latn	34.55	**36.55**	cym_Latn	45.09	**49.54**
dan_Latn	45.89	**51.74**	dzo_Tibt	33.62	**33.99**	ell_Grek	39.16	**43.04**	est_Latn	44.16	**47.71**	eus_Latn	44.93	**48.09**
ewe_Latn	27.1	**27.87**	fin_Latn	44.2	**48.79**	fon_Latn	16.56	**17.63**	glg_Latn	43.55	**48.02**	grn_Latn	30.24	**31.04**
guj_Gujr	37.1	**40.45**	hne_Deva	29.64	**30.44**	hrv_Latn	42.99	**48.09**	ilo_Latn	43.98	**46.68**	ind_Latn	50.07	**55.44**
ita_Latn	44.39	**48.37**	kab_Latn	26.12	**27.1**	kac_Latn	27.85	**29.04**	kan_Knda	40.54	**44.1**	kas_Arab	22.95	**23.67**
kas_Deva	17.15	**17.88**	kat_Geor	41.81	**44.9**	kea_Latn	35.78	**37.61**	kmr_Latn	33.7	**35.35**	kon_Latn	36.24	**37.43**
kor_Hang	25.15	**29.23**	lao_Laoo	38.13	**39.98**	lin_Latn	38.71	**40.32**	lit_Latn	42.62	**47.39**	lmo_Latn	30.0	**31.49**
ltz_Latn	41.44	**45.1**	lug_Latn	34.0	**35.39**	luo_Latn	27.14	**27.33**	lus_Latn	33.3	**34.14**	mag_Deva	32.78	**33.65**
mai_Deva	34.31	**36.05**	mal_Mlym	40.38	**43.88**	mar_Deva	38.07	**40.93**	min_Latn	41.89	**44.06**	mkd_Cyrl	44.65	**48.62**
mlt_Latn	42.62	**47.33**	mya_Mymr	42.56	**44.17**	nld_Latn	44.19	**47.9**	nno_Latn	42.43	**46.45**	nob_Latn	43.35	**46.56**
pbt_Arab	28.45	**30.04**	pol_Latn	40.63	**44.69**	prs_Arab	36.75	**39.93**	quy_Latn	35.37	**35.86**	run_Latn	31.63	**33.21**
rus_Cyrl	43.15	**47.97**	sag_Latn	20.36	**21.52**	san_Deva	27.9	**29.61**	sat_Olck	28.63	**29.95**	scn_Latn	36.77	**39.9**
sin_Sinh	35.75	**38.05**	slk_Latn	40.81	**45.41**	sna_Latn	41.51	**43.93**	snd_Arab	33.16	**36.0**	som_Latn	36.94	**38.57**
spa_Latn	42.38	**46.18**	srd_Latn	40.51	**43.96**	srp_Cyrl	41.7	**47.09**	ssw_Latn	36.89	**38.53**	sun_Latn	41.18	**44.26**
swe_Latn	45.86	**51.3**	swh_Latn	47.58	**50.84**	szl_Latn	34.92	**36.88**	tat_Cyrl	39.43	**42.32**	tgk_Cyrl	38.33	**40.6**
tgl_Latn	49.2	**51.62**	tha_Thai	43.24	**47.0**	tpi_Latn	40.08	**40.7**	twi_Latn	31.6	**32.54**	uig_Arab	37.42	**39.42**
ukr_Cyrl	41.18	**45.68**	urd_Arab	37.15	**40.79**	uzn_Latn	44.17	**46.69**	war_Latn	44.35	**46.44**	zho_Hans	24.44	**30.46**

Table 9: Results on DEEPSEEK-V3 in chrF scores on 100 languages from English into other languages.

Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High
acm_Arab	79.82	**80.02**	ajp_Arab	78.76	**79.16**	arb_Arab	82.95	**85.1**	ars_Arab	82.48	**83.41**	bel_Cyrl	83.02	**84.84**
bul_Cyrl	85.82	**88.27**	cym_Latn	79.78	**82.84**	dan_Latn	84.29	**87.59**	ell_Grek	84.76	**86.84**	est_Latn	87.09	**89.11**
eus_Latn	82.08	**84.12**	fin_Latn	87.54	**90.23**	glg_Latn	81.05	**83.64**	hrv_Latn	86.29	**88.58**	ind_Latn	86.74	**89.0**
ita_Latn	82.92	**85.22**	kor_Hang	86.78	**88.26**	lao_Laoo	80.56	**81.85**	lit_Latn	85.09	**88.26**	mar_Deva	69.13	**71.45**
mkd_Cyrl	83.87	**86.4**	nld_Latn	82.36	**85.33**	pol_Latn	85.3	**87.72**	rus_Cyrl	85.49	**87.73**	san_Deva	70.71	**71.68**
slk_Latn	84.98	**87.64**	snd_Arab	73.78	**75.95**	som_Latn	75.33	**76.9**	spa_Latn	80.78	**83.2**	srp_Cyrl	84.27	**86.98**
swe_Latn	84.61	**87.69**	swh_Latn	79.48	**81.43**	tha_Thai	85.22	**86.79**	uig_Arab	80.28	**81.85**	ukr_Cyrl	85.55	**87.98**
urd_Arab	78.62	**80.58**	uzn_Latn	86.76	**88.09**									

Table 10: Results on DEEPSEEK-V3 in COMET scores on 37 supported languages from English into other languages.

Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High
acm_Arab	2.83	**3.79**	acq_Arab	3.41	**4.43**	aeb_Arab	2.12	**3.19**	ajp_Arab	3.14	**4.28**	als_Latn	8.96	**12.97**
arb_Arab	5.77	**8.92**	ars_Arab	4.76	**6.17**	ary_Arab	1.27	**1.84**	arz_Arab	2.84	**4.6**	awa_Deva	2.23	**3.03**
ayr_Latn	0.52	**0.67**	ban_Latn	2.7	**3.47**	bel_Cyrl	3.07	**4.6**	bho_Deva	2.82	**4.06**	bjn_Latn	2.71	**3.12**
bul_Cyrl	10.5	**16.51**	ceb_Latn	12.3	**16.0**	ckb_Arab	0.2	**0.48**	crh_Latn	0.8	**1.08**	cym_Latn	12.34	**16.61**
dan_Latn	13.1	**19.69**	dzo_Tibt	0.08	**0.08**	ell_Grek	7.42	**11.25**	est_Latn	6.16	**9.59**	eus_Latn	2.49	**4.58**
ewe_Latn	0.61	**0.67**	fin_Latn	5.24	**8.74**	fon_Latn	0.26	**0.4**	glg_Latn	10.71	**15.36**	grn_Latn	1.07	**1.28**
guj_Gujr	2.58	**3.55**	hne_Deva	2.36	**3.3**	hrv_Latn	7.88	**12.22**	ilo_Latn	5.75	**7.02**	ind_Latn	13.96	**19.42**
ita_Latn	9.64	**14.48**	kab_Latn	0.36	**0.45**	kac_Latn	0.39	**0.46**	kan_Knda	1.39	**1.84**	kas_Arab	0.3	**0.35**
kas_Deva	0.09	**0.18**	kat_Geor	2.23	**2.85**	kea_Latn	2.54	**2.83**	kmr_Latn	1.27	**1.67**	kon_Latn	1.35	**1.44**
kor_Hang	3.45	**4.99**	lao_Laoo	**1.07**	0.8	lin_Latn	2.77	**3.43**	lit_Latn	5.93	**9.37**	lmo_Latn	1.75	**2.06**
ltz_Latn	4.46	**6.0**	lug_Latn	1.3	**1.58**	luo_Latn	0.98	**1.25**	lus_Latn	2.1	**2.19**	mag_Deva	3.97	**5.16**
mai_Deva	3.09	**3.75**	mal_Mlym	0.64	**1.0**	mar_Deva	2.45	**3.19**	min_Latn	2.94	**3.51**	mkd_Cyrl	9.08	**12.63**
mlt_Latn	5.7	**8.11**	mya_Mymr	0.21	**0.33**	nld_Latn	8.38	**11.61**	nno_Latn	8.7	**13.66**	nob_Latn	8.77	**13.53**
pbt_Arab	2.14	**3.01**	pol_Latn	6.26	**10.36**	prs_Arab	5.24	**7.12**	quy_Latn	**0.53**	0.51	run_Latn	1.84	**2.0**
rus_Cyrl	8.82	**14.08**	sag_Latn	**0.74**	0.66	san_Deva	**0.22**	0.2	sat_Olck	0.0	**0.15**	scn_Latn	3.05	**4.07**
sin_Sinh	0.66	**1.14**	slk_Latn	8.02	**12.21**	sna_Latn	2.28	**2.77**	snd_Arab	3.82	**5.65**	som_Latn	2.97	**3.94**
spa_Latn	10.31	**13.52**	srd_Latn	3.05	**3.67**	srp_Cyrl	7.26	**12.01**	ssw_Latn	0.78	**0.93**	sun_Latn	4.62	**6.5**
swe_Latn	11.57	**19.5**	swh_Latn	10.94	**14.41**	szl_Latn	2.34	**2.74**	tat_Cyrl	3.45	**5.06**	tgk_Cyrl	3.43	**4.92**
tgl_Latn	15.29	**19.16**	tha_Thai	1.24	**1.61**	tpi_Latn	5.91	**6.67**	twi_Latn	1.87	**2.06**	uig_Arab	0.31	**0.49**
ukr_Cyrl	7.91	**12.1**	urd_Arab	5.38	**8.28**	uzn_Latn	3.76	**4.95**	war_Latn	11.11	**13.62**	zho_Hans	**0.59**	0.33

Table 11: Results on GPT4o-mini in BLEU scores on 100 languages from English into other languages.

Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High
acm_Arab	36.79	**38.99**	acq_Arab	35.92	**38.38**	aeb_Arab	33.7	**36.16**	ajp_Arab	37.53	**40.66**	als_Latn	41.18	**44.51**
arb_Arab	39.4	**43.97**	ars_Arab	38.25	**40.32**	ary_Arab	32.15	**34.47**	arz_Arab	35.72	**38.95**	awa_Deva	29.79	**31.08**
ayr_Latn	25.05	**25.17**	ban_Latn	34.58	**36.19**	bel_Cyrl	34.16	**36.62**	bho_Deva	29.98	**32.09**	bjn_Latn	33.84	**35.02**
bul_Cyrl	43.87	**48.82**	ceb_Latn	45.81	**48.45**	ckb_Arab	27.2	**28.15**	crh_Latn	27.31	**28.67**	cym_Latn	41.43	**45.22**
dan_Latn	45.5	**50.65**	dzo_Tibt	22.06	**22.24**	ell_Grek	38.47	**42.19**	est_Latn	42.39	**46.11**	eus_Latn	41.21	**44.4**
ewe_Latn	**19.29**	19.06	fin_Latn	42.96	**47.34**	fon_Latn	**13.41**	13.24	glg_Latn	42.29	**46.29**	grn_Latn	24.82	**25.43**
guj_Gujr	31.84	**34.51**	hne_Deva	29.56	**30.68**	hrv_Latn	41.98	**46.32**	ilo_Latn	39.6	**42.04**	ind_Latn	49.16	**53.94**
ita_Latn	43.36	**47.28**	kab_Latn	**18.8**	18.53	kac_Latn	**20.99**	20.59	kan_Knda	33.04	**35.91**	kas_Arab	18.55	**18.61**
kas_Deva	15.25	**15.67**	kat_Geor	38.23	**40.15**	kea_Latn	31.77	**32.95**	kmr_Latn	28.39	**28.97**	kon_Latn	**26.63**	26.4
kor_Hang	23.67	**27.85**	lao_Laoo	21.92	**22.96**	lin_Latn	34.6	**35.08**	lit_Latn	41.27	**45.91**	lmo_Latn	28.13	**29.46**
ltz_Latn	37.25	**40.18**	lug_Latn	28.77	**29.4**	luo_Latn	**24.04**	23.58	lus_Latn	28.41	**29.05**	mag_Deva	31.85	**33.36**
mai_Deva	32.64	**34.48**	mal_Mlym	33.35	**35.19**	mar_Deva	35.55	**37.93**	min_Latn	33.55	**35.11**	mkd_Cyrl	42.78	**46.41**
mlt_Latn	39.32	**42.66**	mya_Mymr	33.82	**34.75**	nld_Latn	43.18	**46.85**	nno_Latn	41.2	**45.15**	nob_Latn	42.66	**46.2**
pbt_Arab	28.0	**29.61**	pol_Latn	39.07	**42.88**	prs_Arab	36.32	**39.37**	quy_Latn	**27.19**	27.12	run_Latn	32.32	**33.95**
rus_Cyrl	41.19	**45.64**	sag_Latn	**15.51**	15.31	san_Deva	25.63	**26.42**	sat_Olck	**15.09**	14.84	scn_Latn	33.87	**36.14**
sin_Sinh	26.95	**28.23**	slk_Latn	39.56	**44.11**	sna_Latn	38.08	**39.71**	snd_Arab	32.23	**35.04**	som_Latn	37.31	**39.34**
spa_Latn	41.63	**45.04**	srd_Latn	33.17	**33.97**	srp_Cyrl	39.95	**44.62**	ssw_Latn	31.16	**32.14**	sun_Latn	39.64	**42.83**
swe_Latn	44.77	**50.2**	swh_Latn	46.01	**49.52**	szl_Latn	30.11	**31.39**	tat_Cyrl	37.31	**40.06**	tgk_Cyrl	35.62	**37.94**
tgl_Latn	48.18	**50.73**	tha_Thai	41.28	**44.1**	tpi_Latn	35.17	**35.32**	twi_Latn	27.28	**28.16**	uig_Arab	29.46	**30.7**
ukr_Cyrl	40.25	**44.27**	urd_Arab	35.43	**38.62**	uzn_Latn	42.67	**45.03**	war_Latn	43.92	**46.35**	zho_Hans	22.72	**27.62**

Table 12: Results on GPT-4o-mini in chrF scores on 100 languages from English into other languages.

Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High	Language	Low	High
acm_Arab	**79.87**	79.66	ajp_Arab	77.15	**78.26**	arb_Arab	81.29	**83.8**	ars_Arab	81.23	**81.5**	bel_Cyrl	79.82	**82.32**
bul_Cyrl	83.77	**87.16**	cym_Latn	75.39	**79.75**	dan_Latn	83.2	**86.9**	ell_Grek	83.91	**86.36**	est_Latn	85.22	**87.48**
eus_Latn	77.48	**80.95**	fin_Latn	86.45	**89.23**	glg_Latn	79.21	**82.63**	hrv_Latn	85.04	**87.91**	ind_Latn	85.61	**88.27**
ita_Latn	81.69	**84.54**	kor_Hang	85.16	**87.18**	lao_Laoo	48.57	**51.22**	lit_Latn	84.51	**87.16**	mar_Deva	65.34	**68.35**
mkd_Cyrl	81.63	**84.46**	nld_Latn	81.21	**84.58**	pol_Latn	83.65	**86.67**	rus_Cyrl	83.43	**86.25**	san_Deva	63.69	**64.82**
slk_Latn	83.36	**86.67**	snd_Arab	72.46	**75.13**	som_Latn	76.32	**77.52**	spa_Latn	79.84	**82.72**	srp_Cyrl	81.37	**85.24**
swe_Latn	83.02	**86.87**	swh_Latn	78.6	**80.88**	tha_Thai	83.69	**85.28**	uig_Arab	63.02	**65.16**	ukr_Cyrl	84.11	**86.82**
urd_Arab	77.2	**79.5**	uzn_Latn	84.76	**86.58**									

Table 13: Results on GPT-4o-mini in COMET scores on 37 supported languages from English into other languages.

Models	GPT-4o-mini	DeepSeek-V3	Qwen2.5-14B-Instruct
Tool Selection Accuracy			
Low-frequency partition	0.6053	0.6140	0.6316
High-frequency partition	**0.6667**	**0.6404**	**0.6667**
Accuracy with Correct Tool Using			
Low-frequency partition	0.4386	0.4649	0.4298
High-frequency partition	**0.4912**	**0.4737**	**0.4474**

Table 14: Results reported in accuracy on the partition of TC. We see that the high-frequency partition gives better results on all baseline models.

Models	kea_Latn	kik_Latn	lvs_Latn	pag_Latn
BLEU				
low-frequency	0.9504	0.6983	0.7781	0.9814
high-frequency	**1.1528**	**0.7257**	**1.2053**	**1.0204**
chrF				
low-frequency	28.6936	22.1032	29.0109	29.8830
high-frequency	**29.8472**	**22.9479**	**29.0681**	**30.4843**

Table 15: Results of using low-frequency and high-frequency partitions with fine-tuning models with CTFT on translation from English into other languages. COMET is not reported due to unsupported languages. Results indicate that prompting with higher-frequency paraphrases on the model tuned with CTFT is still useful.

Ground–truth Serbian Cyrillic	Две песме из филма су биле номиноване за најбољу оригиналну песму, а то су „Аудиција“ („Будале које сањају“) и „Град звезда“. Лајонсгејт студио је имао 26 номинација — више него било који други студио.
High–frequency Input	Two tunes from the film, Audition (The Fools Who Dream) and City of Stars, were in the running for best new tune. Lionsgate studio scored 26 nominations — more than everyone else.
High–frequency Output	Два нумера из филма, "Аудиција (Будале које сањају)" и "Град звезда", номинована су за најбољи нови нумер. Студио Лајонсгејт добио је 26 номинација — више од било ког другог студија.
High–frequency Scores	**bleu：0.6189** **chrf：51.7009** **COMET Score: 0.887192964553833**
Low–frequency Input	Two musical selections from the cinematic production, Audition (The Fools Who Dream) and City of Stars, were granted nods for the honor of best original track. Lionsgate production house secured 26 nominations — surpassing all other studios.
Low–frequency Output	ве музичке нумере из филмске продукције, Audition (The Fools Who Dream) и City of Stars, добиле су номинације за награду за најбољу оригиналну песму. Продуцентска кућа Lionsgate обезбедила је 26 номинација — надмашивши све остале студије.
Low–frequency Scores	bleu：0.4717 chrf：36.6703COMET Score: 0.8209505677223206
Original Input	Two songs from the movie, Audition (The Fools Who Dream) and City of Stars, received nominations for best original song. Lionsgate studio received 26 nominations — more than any other studio.
Original Output	Две песме из филма, Audition (The Fools Who Dream) и City of Stars, добиле су номинације за најбољу оригиналну песму. Студио Lionsgate добио је 26 номинација — више од било ког другог студија.
Original Scores	bleu：0.5230 chrf：43.4805 COMET Score: 0.8451238870620728

Ground–truth Serbian Cyrillic	Подивљала деца су можда доживела тешко злостављање у детињству или трауму пре него што су била напуштена или што су побегла.
High–frequency Input	Feral children may have faced severe child abuse or trauma before being deserted or running away.
High–frequency Output	Деца дивљачи можда су се суочила са тешким злостављањем или траумом пре него што су напуштена или побегла.
High–frequency Scores	**bleu：0.6358** **chrf：51.1210** **COMET Score: 0.887217104434967**
Low–frequency Input	Untamed children may have undergone intense child exploitation or mental harm before their forsaking or escape.
Low–frequency Output	Неукрочена деца су можда претрпела интензивну експлоатацију или психичку штету пре њиховог напуштања или бекства.
Low–frequency Scores	bleu：0.4026 chrf：29.9722 COMET Score: 0.7990109324455261
Original Input	Feral children may have experienced severe child abuse or trauma before being abandoned or running away.
Original Output	Деца дивљачи могу да су доживела тешко злостављање или трауму пре напуштања или бега.
Original Scores	bleu：0.5045 chrf：47.2366 COMET Score: 0.7920648455619812

Figure 6: Case studies on translating following our proposed framework. Best results are bolded and highlighted.

Language	Correlation
ilo_Latn	0.9278
srp_Cyrl	0.8950
bho_Deva	0.9506
lao_Laoo	1.0000
mya_Mymr	1.0000
kab_Latn	1.0000
kas_Deva	1.0000

Table 16: The correlation between textual frequency and the final translation BLEU scores on translating from English into other languages. We compute Pearson correlation coefficients (Freedman et al., 2007) using numpy.corrcoef().

Tasks	high-freq	low-freq
Math Reasoning		
#. Total	526	526
0.0-1.5	3	60
1.5-2.5	225	418
2.5-3.5	239	45
3.5-4.5	50	3
4.5-5.5	9	0
Machine Translation		
#. Total	738	738
1.0-1.5	198	73
1.5-2.0	397	402
2.0-2.5	132	216
2.5-3.0	10	41
3.0-3.5	1	6

Table 17: The statistics are based on the TFD calculations: We first statistically calculate the occurrence frequencies of unigrams and bigrams from both the web resources and the generated corpus, then assign different weights to the two corpora, and finally calculate the weighted geometric average of the unigram and bigram frequencies.

Model Size	Low	High
0.5b	0.273	0.325
1.5b	0.442	0.484
3b	0.528	0.581
7b	0.595	0.671
14b	0.600	0.690
32b	0.612	0.680
72b	0.610	0.686

Table 18: The evaluation on different model sizes using qwen-2.5. The results are reported on the task of MR.

Hyperparameter	Value
quantization_bit	4
stage	sft
do_train	true
finetuning_type	lora
lora_target	all
template	qwen
cutoff_len	1024
max_samples	3000
overwrite_cache	true
preprocessing_num_workers	16
logging_steps	10
save_steps	500
per_device_train_batch_size	1
gradient_accumulation_steps	8
learning_rate	1.0e-4
num_train_epochs	10.0
lr_scheduler_type	cosine
warmup_ratio	0.1
bf16	true

Table 19: A list of hyperparameters used in our fine-tuning experiments.

Language Class	Number
0	16
1	46
2	5
3	17
4	12
5	4

Table 20: A list of language classes of the 100 languages used in our experiments. More than half of the languages used in our study are relatively low-resource according to Joshi et al. (2020).

Metrics	Low	High
chrF	18.823	32.873
ROUGE	0.175	0.310
BERTScore	0.492	0.838

Table 21: The evaluation of the chain-of-thought process on the MR partition of our proposed TFPD dataset.

A Scope and Proof Strategy

This document provides a self-contained formal proof for the Textual Frequency Law (TFL). The central claim is:

When two text sequences express the same meaning (i.e., are paraphrases), the one with higher sentence-level frequency tends to incur a lower negative log-likelihood (NLL) loss under a language model trained via cross-entropy minimisation.

The proof proceeds in two parts. Part I (Section D) establishes the relationship between token-level NLL loss and token frequency rank under Zipf's law. Part II (Section E) lifts the token-level result to the sentence level by introducing a sentence-frequency measure and accounting for the gap between marginal and conditional token predictions. Section F discusses the relationship between the mathematical conclusion (loss ordering) and the empirical observation (task performance ordering). Section H catalogues the limitations of the theoretical framework.

Throughout, all logarithms are natural logarithms (base e; units: nats).

B Notation

- V: vocabulary (finite set of tokens).
- w: a token in V; w_r denotes the token with frequency rank r ($r = 1$ is the most frequent).
- $P(w)$: true marginal probability of token w in the training distribution.
- $Q_\theta(w)$: marginal probability assigned to token w by a language model with parameters θ.
- $Q_\theta(w \mid c)$: conditional probability of w given context c under the autoregressive model.
- $\ell^{\mathrm{m}}_\theta(w) \triangleq -\ln Q_\theta(w)$: **marginal** token-level NLL loss.
- $\ell^{\mathrm{c}}_\theta(x_k \mid x_{<k}) \triangleq -\ln Q_\theta(x_k \mid x_1, \ldots, x_{k-1})$: **conditional** token-level NLL loss in an autoregressive model.
- $x = (x_1, x_2, \ldots, x_K)$: a sentence (token sequence) of length K.
- $\ell_\theta(x) \triangleq \frac{1}{K}\sum_{k=1}^{K} \ell^{\mathrm{c}}_\theta(x_k \mid x_{<k})$: average conditional NLL loss of sentence x — the quantity the autoregressive model actually computes.
- $\mathrm{sfreq}(x)$: sentence-level frequency, defined in Assumption 4.
- $Z = \sum_{n=1}^{|V|} n^{-s}$: Zipf normalisation constant; $C \triangleq \ln Z > 0$.

Remark 1 (Marginal vs. conditional loss). It is essential to distinguish the marginal loss $\ell^{\mathrm{m}}_\theta(w)$ from the conditional loss $\ell^{\mathrm{c}}_\theta(x_k \mid x_{<k})$. The Zipf-based analysis in Part I operates on marginal quantities. Part II bridges to the conditional quantities that autoregressive models actually use, via an explicit error term.

C Assumptions

We state four formal assumptions that the proof depends on, followed by one contextual remark on the training objective.

Assumption 1 (Zipf's Law for Token Frequencies)**.** The true marginal probability of token w_r with rank r satisfies

$$P(w_r) = \frac{r^{-s}}{Z}, \quad s > 0, \quad Z = \sum_{n=1}^{|V|} n^{-s}.$$

Zipf's law is a well-documented empirical regularity for the marginal frequency of tokens aggregated over a large corpus. It characterises the bulk of the vocabulary distribution accurately, though deviations occur in the extreme tail (very rare tokens). We treat s as a fixed positive constant.

Assumption 2 (Rank-Dependent Log-Domain Approximation)**.** After training, for every token $w_r \in V$ there exists a rank-dependent bound $\varepsilon(r) \geq 0$ such that

$$\left|\ln Q_\theta(w_r) - \ln P(w_r)\right| \leq \varepsilon(r). \tag{7}$$

Remark 2 (Strength and character of Assumption 2). Equation (7) is equivalent to a multiplicative approximation guarantee:

$$e^{-\varepsilon(r)} \leq \frac{Q_\theta(w_r)}{P(w_r)} \leq e^{\varepsilon(r)}, \quad \forall\, r.$$

This is a *pointwise* condition on every token — considerably stronger than merely controlling the expected cross-entropy loss. Standard cross-entropy training minimises $\mathbb{E}_{w\sim P}[-\ln Q_\theta(w)]$, which controls the P-weighted average loss but does not, by itself, guarantee pointwise log-domain accuracy for each individual token.

We expect $\varepsilon(r)$ to be small for high-frequency tokens (small r), because these tokens are observed abundantly during training and the model receives strong gradient signal for them. For low-frequency tokens (large r), the model may see very few training examples, and $\varepsilon(r)$ is expected to grow. All subsequent results are stated in terms of $\varepsilon(r)$, so the reader can assess the strength of each conclusion as a function of the model's approximation quality at each frequency tier.

Assumption 2 is **not derivable** from the training objective alone. It is an empirical hypothesis about the outcome of training — motivated by the fact that cross-entropy minimisation encourages $Q_\theta \to P$, but not logically entailed by it.

Empirical motivation. Although no existing study directly measures the pointwise bound $\varepsilon(r)$ as a function of rank, several independent lines of evidence support the plausibility of Assumption 2:

(a) *LLM token distributions follow Zipf's law.* Mikhaylovskiy (2025) shows that text generated by large language models obeys Zipf's law, though the fit quality depends on decoding temperature. This indicates that the model's output distribution Q_θ preserves the rank–frequency structure of the training distribution P, a necessary (though not sufficient) condition for small $\varepsilon(r)$.

(b) *LLMs encode token frequency in their prediction heads.* Kobayashi et al. (2023) demonstrate that the bias terms in the prediction head of Transformer language models (BERT and GPT-2) significantly reflect corpus word frequency, effectively encoding a frequency prior consistent with logit adjustment in long-tail learning. This suggests that the model's internal mechanism is structured in a way that facilitates accurate frequency-based predictions.

(c) *Frequency modulates model–human surprisal alignment.* Oh et al. (2024) find that word frequency systematically modulates the gap between LLM surprisal estimates and human reading times, with larger models predicting low-frequency words "too accurately" relative to human expectations. This is consistent with the view that well-trained models achieve small $\varepsilon(r)$ for high-frequency tokens and progressively larger errors in the tail.

(d) *Downstream performance correlates with Zipfian fit.* He et al. (2025) show that pre-trained models consistently achieve optimal downstream performance when the vocabulary size is chosen so that the resulting token frequency distribution follows Zipf's law. Their experiments across NLP, genomics, and chemistry establish a link between Zipfian alignment at the tokenisation level and model quality, reinforcing the broader premise that power-law regularity in the token distribution — a key ingredient of Assumption 2 — is conducive to effective language modelling.

These findings collectively support the hypothesis that $\varepsilon(r)$ is small for high-frequency tokens and grows with rank, but a direct empirical characterisation of the pointwise bound remains an open problem.

Assumption 3 (Bounded Marginal–Conditional Discrepancy)**.** For each token x_k in a sentence $x = (x_1, \ldots, x_K)$, define the **contextual discrepancy**:

$$\eta_{x_k} \triangleq \ell_\theta^{\mathrm{c}}(x_k \mid x_{<k}) - \ell_\theta^{\mathrm{m}}(x_k) = \ln Q_\theta(x_k) - \ln Q_\theta(x_k \mid x_1, \ldots, x_{k-1}).$$

We assume that for each sentence x, the average contextual discrepancy is bounded:

$$|\bar{\eta}_x| \leq \eta_x, \quad \text{where} \quad \bar{\eta}_x \triangleq \frac{1}{K}\sum_{k=1}^{K} \eta_{x_k},$$

and $\eta_x \geq 0$ is a sentence-dependent bound.

Remark 3 (Nature of η_{x_k}). The sign and magnitude of η_{x_k} depend on how informative the context $x_{<k}$ is for predicting x_k:

- $\eta_{x_k} < 0$: the context makes x_k *more* predictable than its marginal frequency suggests (conditional probability exceeds marginal). This is typical for high-frequency function words in predictable contexts (e.g., "of" after "United States").
- $\eta_{x_k} > 0$: the context makes x_k *less* predictable (e.g., a token that is common in isolation but surprising in the given context).
- $\eta_{x_k} \approx 0$: the context is approximately uninformative for x_k.

For sentences composed of common, high-frequency tokens (the "high-frequency paraphrases" central to TFL), many constituent tokens have highly predictable collocations, so η_{x_k} tends to be negative. This directional tendency is *favourable* to TFL: it means the actual conditional loss is systematically lower than the marginal-based estimate for high-frequency sentences. However, we do not rely on this tendency in the proof; instead, we use the conservative absolute bound $|\bar{\eta}_x| \leq \eta_x$.

Note that η_x is **sentence-dependent**: different sentences may have different bounds. We do not assume a single universal bound across all sentences.

Assumption 4 (Sentence Frequency via Geometric Mean of Token Frequencies)**.** The sentence-level frequency of $x = (x_1, \ldots, x_K)$ is defined as

$$\mathrm{sfreq}(x) \triangleq \Big(\prod_{k=1}^{K} P(x_k)\Big)^{1/K},$$

or equivalently in log-space:

$$\ln \mathrm{sfreq}(x) = \frac{1}{K}\sum_{k=1}^{K} \ln P(x_k). \tag{8}$$

This definition treats sentence frequency as the geometric mean of marginal token frequencies, corresponding to a unigram model for the sentence probability. It ignores word order and inter-token dependencies, which is a deliberate simplification: the goal is a tractable frequency measure that correlates with how "common" the constituent vocabulary of a sentence is. For comparing paraphrases with identical meaning but different word choices, this measure captures precisely the relevant variation — the frequency tier of the vocabulary used.

Remark 4 (Role of the training objective). Standard language model training minimises the expected negative log-likelihood: $\min_\theta \mathbb{E}_{w\sim P}[-\ln Q_\theta(w)]$. This training objective *motivates* Assumption 2: under ideal conditions with sufficient capacity and data, the minimiser satisfies $Q_\theta(w) = P(w)$ for all w, which would give $\varepsilon(r) = 0$ everywhere. In practice, finite data and model capacity lead to nonzero $\varepsilon(r)$, particularly for low-frequency tokens. The training objective does **not** appear as a formal assumption because the proof does not directly invoke it; it serves as the background justification for why Assumption 2 is plausible.

D Part I: Token-Level Results

D.1 Step 1: Self-Information under Zipf's Law

By Assumption 1, the self-information (ideal NLL) of token w_r is

$$\begin{aligned} -\ln P(w_r) &= -\ln\left(\frac{r^{-s}}{Z}\right) \\ &= -(-s\ln r - \ln Z) \\ &= s\ln r + \ln Z. \end{aligned} \tag{9}$$

Setting $C \triangleq \ln Z > 0$:

$$-\ln P(w_r) = s\ln r + C. \tag{10}$$

This shows that the ideal NLL is *affine* in $\ln r$ with slope s and intercept C.

D.2 Step 2: Model Loss Bounded by Approximation Error

By Assumption 2:

$$-\varepsilon(r) \le \ln Q_\theta(w_r) - \ln P(w_r) \le \varepsilon(r).$$

Multiplying through by -1 (which reverses the inequalities):

$$-\ln P(w_r) - \varepsilon(r) \le -\ln Q_\theta(w_r) \le -\ln P(w_r) + \varepsilon(r).$$

Defining $\ell_\theta^{\mathrm{m}}(w_r) \triangleq -\ln Q_\theta(w_r)$, we can write:

$$\ell_\theta^{\mathrm{m}}(w_r) = -\ln P(w_r) + \delta_{w_r}, \qquad |\delta_{w_r}| \le \varepsilon(r), \tag{11}$$

where $\delta_{w_r} \triangleq -\ln Q_\theta(w_r) - (-\ln P(w_r)) = \ln P(w_r) - \ln Q_\theta(w_r)$ is the signed approximation error for token w_r.

D.3 Step 3: Semi-Log Linear Relationship

Substituting (10) into (11):

$$\ell_\theta^{\mathrm{m}}(w_r) = s\ln r + C + \delta_{w_r}, \qquad |\delta_{w_r}| \le \varepsilon(r). \tag{12}$$

Theorem 1 (Token-Level Semi-Log Linearity)**.** *Under Assumptions 1 and 2, the marginal token-level NLL loss satisfies*

$$\ell_\theta^{\mathrm{m}}(w_r) = s\ln r + C + \delta_{w_r}, \quad |\delta_{w_r}| \le \varepsilon(r),$$

where $s > 0$ is the Zipf exponent and $C = \ln Z > 0$. In the semi-log plane (x-axis: $\ln r$; y-axis: ℓ_θ^{m}), the relationship is linear with slope s and intercept C, within a rank-dependent error band of half-width $\varepsilon(r)$.

Proof. Immediate from the chain of equalities in Steps 1–3. □

Remark 5 (Semi-log vs. log-log). Equation (12) is a *semi-log* linear relationship (ℓ_θ^{m} is affine in $\ln r$), **not** a log-log relationship (which would require $\ln \ell_\theta^{\mathrm{m}}$ to be affine in $\ln r$, i.e., a power law for the loss itself).

D.4 Token-Level Monotonicity

Theorem 2 (Sufficient Condition for Strict Token-Level Monotonicity)**.** *Let w_i, w_j be two tokens with $r_i < r_j$ (i.e., $P(w_i) > P(w_j)$). A sufficient condition for $\ell_\theta^{\mathrm{m}}(w_i) < \ell_\theta^{\mathrm{m}}(w_j)$ is*

$$\varepsilon(r_i) + \varepsilon(r_j) < s\ln\left(\frac{r_j}{r_i}\right). \tag{13}$$

In the special case of a uniform bound $\varepsilon(r) \equiv \varepsilon$, this reduces to

$$\frac{r_j}{r_i} > e^{2\varepsilon/s}. \tag{14}$$

Proof. We require the worst-case upper bound of $\ell_\theta^{\mathrm{m}}(w_i)$ to be strictly less than the worst-case lower bound of $\ell_\theta^{\mathrm{m}}(w_j)$:

$$\big(s \ln r_i + C + \varepsilon(r_i)\big) < \big(s \ln r_j + C - \varepsilon(r_j)\big).$$

Cancelling C and rearranging:

$$\varepsilon(r_i) + \varepsilon(r_j) < s(\ln r_j - \ln r_i) = s \ln\left(\frac{r_j}{r_i}\right).$$

When $\varepsilon(r) \equiv \varepsilon$, this becomes $2\varepsilon < s \ln(r_j/r_i)$, i.e., $r_j/r_i > e^{2\varepsilon/s}$. □

Remark 6 (When monotonicity fails). For adjacent-rank tokens ($r_j = r_i + 1$), the rank ratio is $1 + 1/r_i \to 1$ as $r_i \to \infty$, so the left-hand side of (13) approaches zero while the right-hand side remains positive but also approaches zero (as $\ln(1 + 1/r_i) \approx 1/r_i$). Condition (13) fails whenever the approximation error exceeds the Zipf-induced gap. Strict ordering between tokens of similar frequency **cannot be guaranteed** in the tail of the distribution. This is an inherent limitation: cross-entropy training provides diminishing approximation quality for rarer tokens.

E Part II: Sentence-Level Extension

This part bridges the token-level results to the sentence level.

E.1 Setup

Let $x = (x_1, \ldots, x_K)$ and $x' = (x'_1, \ldots, x'_{K'})$ be two sentences. Their sentence-level losses (as computed by an autoregressive model) are

$$\ell_\theta(x) = \frac{1}{K} \sum_{k=1}^{K} \ell_\theta^{\mathrm{c}}(x_k \mid x_{<k}), \tag{15}$$

$$\ell_\theta(x') = \frac{1}{K'} \sum_{k=1}^{K'} \ell_\theta^{\mathrm{c}}(x'_k \mid x'_{<k}). \tag{16}$$

Their log sentence-frequencies under Assumption 4 are

$$\ln \mathrm{sfreq}(x) = \frac{1}{K} \sum_{k=1}^{K} \ln P(x_k), \qquad \ln \mathrm{sfreq}(x') = \frac{1}{K'} \sum_{k=1}^{K'} \ln P(x'_k).$$

Note that $-\ln \mathrm{sfreq}(x) = \frac{1}{K} \sum_{k=1}^{K} (-\ln P(x_k))$, i.e., the negative log sentence-frequency equals the average ideal marginal NLL.

E.2 Step 4: Decomposing Sentence-Level Loss

For each token x_k with rank r_k, the conditional loss can be decomposed as follows:

$$\ell_\theta^{\mathrm{c}}(x_k \mid x_{<k}) = \underbrace{-\ln P(x_k)}_{\text{ideal marginal NLL}} + \underbrace{\delta_{x_k}}_{\text{marginal approx. error}} + \underbrace{\eta_{x_k}}_{\text{contextual discrepancy}}, \tag{17}$$

where:

- $\delta_{x_k} = \ell_\theta^{\mathrm{m}}(x_k) - (-\ln P(x_k)) = \ln P(x_k) - \ln Q_\theta(x_k)$, with $|\delta_{x_k}| \le \varepsilon(r_k)$ by Assumption 2;
- $\eta_{x_k} = \ell_\theta^{\mathrm{c}}(x_k \mid x_{<k}) - \ell_\theta^{\mathrm{m}}(x_k) = \ln Q_\theta(x_k) - \ln Q_\theta(x_k \mid x_{<k})$, the contextual discrepancy from Assumption 3.

Verification. Adding the three terms on the right-hand side of (17):

$$\begin{aligned} &-\ln P(x_k) + [\ln P(x_k) - \ln Q_\theta(x_k)] + [\ln Q_\theta(x_k) - \ln Q_\theta(x_k \mid x_{<k})] \\ &\qquad = -\ln Q_\theta(x_k \mid x_{<k}) = \ell_\theta^{\mathrm{c}}(x_k \mid x_{<k}). \quad \checkmark \end{aligned} \tag{18}$$

Averaging (17) over all tokens in x:

$$\begin{aligned} \ell_\theta(x) &= \frac{1}{K}\sum_{k=1}^{K} \ell_\theta^{\mathrm{c}}(x_k \mid x_{<k}) \\ &= \underbrace{\frac{1}{K}\sum_{k=1}^{K}(-\ln P(x_k))}_{=-\ln \mathrm{sfreq}(x)} + \underbrace{\frac{1}{K}\sum_{k=1}^{K}\delta_{x_k}}_{\bar{\delta}_x} + \underbrace{\frac{1}{K}\sum_{k=1}^{K}\eta_{x_k}}_{\bar{\eta}_x}. \end{aligned} \tag{19}$$

Define the average marginal approximation bound:

$$\bar{\varepsilon}_x \triangleq \frac{1}{K}\sum_{k=1}^{K}\varepsilon(r_k).$$

By the triangle inequality, $|\bar{\delta}_x| \le \bar{\varepsilon}_x$. By Assumption 3, $|\bar{\eta}_x| \le \eta_x$.

Therefore:

$$\boxed{\ell_\theta(x) = -\ln \mathrm{sfreq}(x) + \bar{\delta}_x + \bar{\eta}_x, \quad |\bar{\delta}_x| \le \bar{\varepsilon}_x, \quad |\bar{\eta}_x| \le \eta_x.} \tag{20}$$

Remark 7 (Tightness of the bound after averaging). The bound $|\bar{\delta}_x| \le \bar{\varepsilon}_x$ is worst-case (triangle inequality). If the token-level errors δ_{x_k} have approximately zero mean and are weakly correlated across positions, a central-limit-type argument gives the tighter practical estimate $|\bar{\delta}_x| \approx O(\bar{\varepsilon}_x/\sqrt{K})$. Similarly for $\bar{\eta}_x$. Thus the sufficient conditions derived below are conservative; in practice, the effective threshold for the TFL to hold is likely smaller by a factor on the order of $1/\sqrt{K}$.

E.3 Sentence-Level Results

Theorem 3 (Sentence-Level Loss–Frequency Relationship)**.** *Under Assumptions 1, 2, 3, and 4, the sentence-level NLL loss satisfies*

$$\ell_\theta(x) = -\ln \mathrm{sfreq}(x) + \bar{\delta}_x + \bar{\eta}_x,$$

with $|\bar{\delta}_x + \bar{\eta}_x| \le \bar{\varepsilon}_x + \eta_x$. That is, the sentence-level loss is approximately equal to the negative log sentence-frequency, up to a total error bounded by $\bar{\varepsilon}_x + \eta_x$.

Proof. Equation (19) gives the exact decomposition. By the triangle inequality:

$$|\bar{\delta}_x + \bar{\eta}_x| \le |\bar{\delta}_x| + |\bar{\eta}_x| \le \bar{\varepsilon}_x + \eta_x. \qquad \square$$

Theorem 4 (Textual Frequency Law — Sufficient Condition)**.** *Let x and x' be two paraphrases with $\mathrm{sfreq}(x) > \mathrm{sfreq}(x')$. A sufficient condition for $\ell_\theta(x) < \ell_\theta(x')$ is*

$$\ln \frac{\mathrm{sfreq}(x)}{\mathrm{sfreq}(x')} > (\bar{\varepsilon}_x + \eta_x) + (\bar{\varepsilon}_{x'} + \eta_{x'}), \tag{21}$$

where $\bar{\varepsilon}_x, \eta_x$ and $\bar{\varepsilon}_{x'}, \eta_{x'}$ are the approximation and contextual error bounds for x and x', respectively.

Proof. By Theorem 3, the worst-case upper bound on $\ell_\theta(x)$ and worst-case lower bound on $\ell_\theta(x')$ are:

$$\begin{aligned} \ell_\theta(x) &\le -\ln \mathrm{sfreq}(x) + (\bar{\varepsilon}_x + \eta_x), \\ \ell_\theta(x') &\ge -\ln \mathrm{sfreq}(x') - (\bar{\varepsilon}_{x'} + \eta_{x'}). \end{aligned}$$

It suffices to require the upper bound on $\ell_\theta(x)$ to be strictly less than the lower bound on $\ell_\theta(x')$:

$$-\ln \text{sfreq}(x) + (\bar{\varepsilon}_x + \eta_x) < -\ln \text{sfreq}(x') - (\bar{\varepsilon}_{x'} + \eta_{x'}).$$

Rearranging (add $\ln \text{sfreq}(x)$ and $(\bar{\varepsilon}_{x'} + \eta_{x'})$ to both sides):

$$(\bar{\varepsilon}_x + \eta_x) + (\bar{\varepsilon}_{x'} + \eta_{x'}) < \ln \text{sfreq}(x) - \ln \text{sfreq}(x') = \ln \frac{\text{sfreq}(x)}{\text{sfreq}(x')},$$

which is precisely condition (21). □

Remark 8 (Sufficient, not necessary). Condition (21) is a sufficient condition. The TFL may hold even when this condition is not met, because:

(i) The worst-case bounds are conservative — actual errors may partially cancel rather than compound.

(ii) The averaging effect across K tokens (Remark 7) typically yields a much tighter effective error, on the order of $(\bar{\varepsilon}_x + \eta_x)/\sqrt{K}$.

(iii) For high-frequency paraphrases, the contextual discrepancy $\bar{\eta}_x$ tends to be negative (Remark 3), which further reduces the actual sentence loss below the worst-case bound.

Remark 9 (Practical magnitude of the condition). The condition requires the log frequency ratio of the two paraphrases to exceed the sum of all error bounds. In practice, paraphrases constructed by substituting a few content words (e.g., "deserted" → "abandoned") while sharing most function words ("the", "was", "in") differ modestly in sentence frequency. Whether (21) is satisfied depends on:

- How many tokens differ, and how large the frequency gap is for those tokens.
- The model's approximation quality ($\varepsilon(r)$) at the relevant frequency tiers.
- The magnitude of the marginal–conditional discrepancy (η).

The theorem provides the analytical framework; the empirical validation in the main paper demonstrates that the TFL holds in practice across a wide range of settings, suggesting that the error terms are typically small enough for the condition to be effectively met.

F Discussion: From Loss Ordering to Task Performance

Theorems 3 and 4 establish that, under the stated assumptions, higher-frequency paraphrases incur lower NLL loss. The empirical claim of the Textual Frequency Law is stronger: higher-frequency paraphrases lead to better *task performance* (e.g., higher accuracy in math reasoning, higher BLEU/chrF in machine translation). Bridging this gap requires additional reasoning that we outline here.

For prompting. When an LLM is prompted with input x, the model generates output $y = (y_1, \ldots, y_T)$ by sampling from or maximising the conditional distribution $Q_\theta(y \mid x)$. Lower NLL loss on x means the model assigns higher probability to the token sequence x. This implies that x falls in a region of the input space where the model's internal representations are better calibrated — having been shaped by more training examples with similar token distributions. An input that the model "understands" better (assigns higher probability to) is more likely to activate the correct reasoning pathways and produce accurate outputs. This argument is plausible and consistent with the empirical evidence, but it is not a formal proof: the relationship between input perplexity and output quality depends on the model's internal mechanism, which is not captured by our framework.

For fine-tuning. In fine-tuning, the model optimises $\sum_n \log Q_\theta(y_n \mid x_n)$ over training pairs (x_n, y_n). If the model already assigns higher probability to the input tokens of high-frequency paraphrases, the gradient signal from these examples is more stable and the effective learning rate for the output mapping is higher. Additionally, high-frequency inputs are closer to the pre-training distribution, reducing the risk of catastrophic forgetting.

Status of this argument. The connection from loss ordering to task performance is an **empirically motivated hypothesis**, not a theorem. The formal contribution of this proof is the loss ordering result (Theorem 4). The task performance connection is supported by extensive experiments in the main paper.

G Summary of Results

Result	Equation	Assumptions Used
Token semi-log linearity (Thm. 1)	(12)	1, 2
Token strict monotonicity (Thm. 2)	(13)	1, 2
Sentence loss–frequency (Thm. 3)	(20)	1, 2, 3, 4
TFL sufficient condition (Thm. 4)	(21)	1, 2, 3, 4

H Limitations

We catalogue the limitations of the theoretical framework for full transparency.

1. **Assumption 2 is not derivable from the training objective.** The pointwise log-domain approximation guarantee is stronger than what cross-entropy minimisation alone can ensure. Cross-entropy training controls the P-weighted expected loss, not the per-token log-domain error. The assumption is empirically motivated but remains a hypothesis about the outcome of training. For low-frequency tokens, $\varepsilon(r)$ may be large, and the theorem's guarantees weaken accordingly. While several studies provide indirect support for the plausibility of this assumption (see Remark 2), a direct empirical measurement of the pointwise bound $\varepsilon(r)$ as a function of rank remains an open problem in the literature.

2. **Contextual discrepancy η_x is difficult to estimate.** The magnitude of η_{x_k} depends on the specific sentence context and the model's learned conditional distributions. No general data-independent bound is available. In the proof, η_x is treated as an axiomatically bounded quantity. Empirically, one could estimate η_x by comparing marginal and conditional perplexities on a held-out corpus, but such estimates would be model- and data-specific.

3. **The sentence frequency measure is a unigram approximation.** The geometric-mean definition (Assumption 4) ignores word order and inter-token dependencies. For paraphrase pairs that differ mainly in word choice (not syntactic structure), this is a reasonable proxy. For paraphrases with substantially different syntactic structures or lengths, the measure may not fully capture the relevant notion of "commonness."

4. **Sentence length differences.** When two paraphrases have different lengths $K \neq K'$, the averaging effect differs: a longer sentence averages over more tokens, which may tighten or loosen the effective error bounds. This interaction is not explicitly modelled; the theorem treats $\bar{\varepsilon}_x$ and η_x as given quantities.

5. **Loss ordering does not formally imply task performance ordering.** The proven result is $\ell_\theta(x) < \ell_\theta(x')$ (lower NLL loss for higher-frequency paraphrases). The claim that this translates to better downstream task performance (higher accuracy, higher BLEU) is empirically supported but not formally established within this framework. See Section F for further discussion.

6. **Semantic equivalence is assumed, not verified.** The TFL compares paraphrases with "the same meaning." The proof assumes perfect semantic equivalence; in practice, paraphrasing inevitably introduces subtle meaning shifts. A formal treatment would require a semantic similarity metric, which is beyond the scope of a frequency-based theorem.

7. **Zipf's law is approximate in the tail.** The power-law model fits well for the bulk of the vocabulary but may deviate for extremely rare tokens. Such deviations are absorbed into $\varepsilon(r)$ in the analysis, but this means the error bound for tail tokens reflects both the model's approximation error and the inadequacy of the Zipf model itself.

I Conclusion

This document has established, under clearly stated assumptions, that:

(i) Token-level NLL loss is semi-log linear in frequency rank (Theorem 1).

(ii) Sentence-level NLL loss is approximately equal to the negative log sentence-frequency, with a bounded error term (Theorem 3).

(iii) When the sentence-frequency ratio between two paraphrases is sufficiently large relative to the error bounds, the higher-frequency paraphrase provably has lower model loss (Theorem 4).

These results provide the theoretical foundation for the Textual Frequency Law. The sufficient condition is conservative; empirical evidence in the main paper demonstrates that the TFL holds broadly in practice, consistent with the error terms being small enough for the condition to be effectively satisfied in typical settings.