Are Generative Models Underconfident? Better Quality Estimation with Boosted Model Probability

Tu Anh Dinh and **Jan Niehues**
Karlsruhe Institute of Technology
Karlsruhe, Germany
{firstname}.{lastname}@kit.edu

Abstract

Quality Estimation (QE) is estimating the quality of the model output during inference when the ground truth is not available. Deriving output quality from the models' output probability is the most trivial and low-effort way. However, we show that the output probability of text-generation models can appear **underconfident**. At each output step, there can be multiple correct options, making the probability distribution spread out more. Thus, lower probability does not necessarily mean lower output quality. Due to this observation, we propose a QE approach called **BOOSTEDPROB**[1], which boosts the model's confidence in cases where there are multiple viable output options. With no increase in complexity, BOOSTEDPROB is notably better than raw model probability in different settings, achieving on average +0.194 improvement in Pearson correlation to ground-truth quality. It also comes close to or outperforms more costly approaches like supervised or ensemble-based QE in certain settings.

1 Introduction

Text generation models, such as transcription and translation systems like Whisper (Radford et al., 2023) or Large Language Models like Llama (Touvron et al., 2023), have demonstrated remarkable effectiveness across various applications (Amorese et al., 2023; Xie et al., 2024; Masalkhi et al., 2024). However, these models could still make mistakes in certain cases, such as when the input is noisy or when the context involves ambiguous phrasing or domain-specific jargon (Katkov et al., 2024; Huang et al., 2023). Consequently, it is crucial to inform users about the reliability of model outputs by offering a quality assessment. This task is formally recognized as Quality Estimation.

Particularly, Quality Estimation (QE) is the task of providing quality scores of model outputs during inference when the ground truth is not available. The most straightforward way is to utilize the model's output probability. While previous works have shown that model probability is prone to be overconfident (Nguyen et al., 2015; Li et al., 2021), in this work, we point out another issue. We show that the output probability on free-form text generation tasks, such as translation or summarization, can be **underconfident**. Specifically, lower probability does not necessarily indicate lower output quality, but could mean that the probability distribution is spread out over multiple correct options.

We propose a simple QE approach, **BOOSTEDPROB**, which only utilizes the model output probability distribution. BOOSTEDPROB tackles the underconfidence phenomenon mentioned above by boosting the model's confidence scores when there are potentially multiple correct output options.

Specifically, our contributions are as follows:

1. We show that, for models performing free-form text generation tasks, at an output step, there can be multiple valid outputs, leading to multiple tokens having dominant mass in the probability distribution. Probability mass spread over these valid tokens makes the model appear **underconfident**.

2. We propose a QE approach, **BOOSTEDPROB**, that boosts the confidence of these dominant tokens. BOOSTEDPROB is easy to implement and does not add any complexity compared to raw model probabilities. It is substantially more efficient than ensemble-based QE, which requires generating multiple outputs, and supervised QE, which is data-dependent and not available for tasks other than translation.

3. We show that BOOSTEDPROB is: (1) notably better as a quality estimator than the raw probabilities across different tasks and models; (2)

[1]Implementation available at https://github.com/TuAnh23/boostedprob.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 3365–3383
November 4-9, 2025

coming close to or outperforming more expensive supervised and ensemble-based baselines in certain settings; (3) with BOOSTEDPROB, improving models' quality comes with improving their self-evaluation ability.

2 Related Work

2.1 Quality Estimation

Model probability is the most trivial estimator of the output quality. However, previous works have shown that using the probability of the final output alone is not optimal, as neural models tend to be overconfident (Nguyen et al., 2015; Li et al., 2021). Another approach is to use the entropy of the whole probability distribution (Fomicheva et al., 2020). However, it does not consider which option is selected in the end. These methods are generally low-effort, with the only drawback that output probability might not be accessible for API-only models. Therefore, probability-based QE has been employed in many use cases, such as for deciding whether to ask users to repeat themselves in dialog systems (Jurafsky and Martin, 2025), or determining the exit layer in early exiting models (Teerapittayanon et al., 2016; Xin et al., 2020).

Other types of QE are usually more costly. Some approaches require generating multiple outputs, such as ensemble-based approaches like Monte Carlo sequence entropy (Malinin and Gales, 2021; Kuhn et al., 2023), Perturbation-based QE (Dinh and Niehues, 2023), and self-validation approaches (Kadavath et al., 2022). Some approaches require access to the model training data to detect out-of-distribution instances during inference (Lee et al., 2018; Ren et al., 2023). Other approaches require an external model to measure the output quality. Prism (Thompson and Post, 2020) uses a multilingual Machine Translation model to score output from other models by forced decoding. Cohen et al. (2023) uses an examiner model to ask questions and discover inconsistencies of the evaluated model.

One outstanding case of external QE modules is supervised QE models for Machine Translation (MT), such as CometKiwi (Rei et al., 2022). For MT, there exists abundant data of *(source, model translation, human-labeled scores)* tuples, which enables training supervised QE models. One can try to avoid the use of costly human-labeled scores by training QE models on synthetic data with synthetic errors (Tuan et al., 2021), or synthetic scores using reference-based metrics (Zouhar et al., 2023) like BLEU (Papineni et al., 2002), chrF (Popović, 2015) or BERTScore (Zhang et al., 2019). Supervised QE has been widely adopted in MT, and is getting close to the performance of reference-based metrics (Freitag et al., 2022).

Nevertheless, supervised approaches are data-dependent, and mostly not available for tasks other than translation. Thus, we focus on using model probability as a quality estimator, given its simplicity and efficiency. Previous works mostly focus on the overconfidence problem of model probability, where one solution is to use larger models with more training data (Naganuma et al., 2025; Chhikara, 2025). We identify another weakness of model probability - being **underconfident** for free-form text generation tasks, and propose a simple modification to the probability to tackle this.

2.2 Dominant Tokens

Previous works have considered that there can be multiple tokens with dominant probability mass in the output distribution. For example, Ott et al. (2018) shows that, for MT, model distribution is highly spread in the hypothesis space. However, they focus on its effect on model fitting and inference search rather than on QE. Other works focus on sampling, where they try to find the set of dominant tokens to sample from during output generation to maintain high quality but also high diversity. Popular sampling strategies includes top-k (Fan et al., 2018), top-p (Holtzman et al.), ϵ-cut (Hewitt et al., 2022), η-cut (Hewitt et al., 2022) and min-p (Nguyen et al., 2024). For top-k, the assumption is that, the top k tokens with the highest probability are the most important ones. For top-p, the most important tokens are those with top probabilities that sum up to p. For ϵ-cut, the most important token probabilities are larger than ϵ. For η-cut, the most important token probabilities are larger than either η or $\sqrt{\eta} \times \exp(-entropy(\mathbb{P}))$, where $\mathbb{P}$ is the output probability distribution. For min-p, the most important tokens have probabilities larger than the top-1 probability multiplied by p.

In our work, we focus on finding dominant tokens to boost their confidence for QE, rather than to support sampling or search strategies during inference.

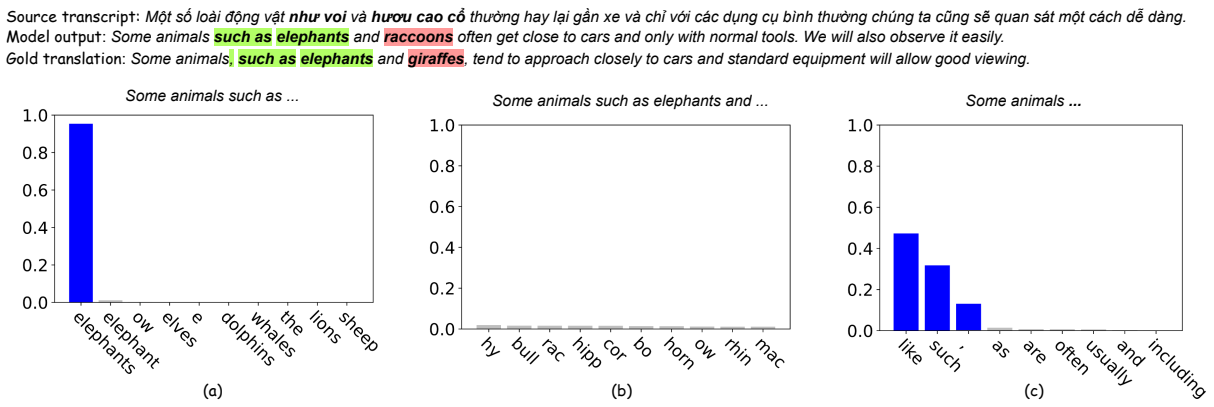

Figure 1: Whisper's output probability distributions. (a) The model gives high probability to the correct translation ("*elephants*"). (b) The model gives low probability to all tokens, and outputs the wrong translation in the end ("*raccoons*" instead of "*giraffes*"). (c) The probabilities are lower due to probability mass being spread out between multiple correct options (",", "*like*" and "*such as*), and ***do not indicate lower quality***.

3 Model Probability – An Underconfident Quality Estimator?

Illustrative example Our investigation begins with the example in Figure 1, where Whisper Large V3 (Radford et al., 2023) translates a Vietnamese audio sentence into English. Figure 1a (correct translation to "*elephants*" and Figure 1b (wrong translation to "*raccoons*") are intuitive: higher probabilities indicate better output quality. However, in Figure 1c, most probability mass is spread between three options: the comma ",", "*like*" and "*such as*", all of which are reasonable outputs. The probabilities here are lower, but do not indicate low output quality. We suspect this happens due to the ambiguous nature of the Speech Translation task.

Ambiguous Tasks By "ambiguous tasks", we refer to tasks where for an input, there can be multiple valid output options. We investigate the model behaviors when working on text-generation tasks with ambiguity like Speech Translation (ST), where for an input, multiple translations can be valid. We do so by comparing to the less ambiguous Automatic Speech Recognition (ASR) task, where for an input audio, there is only one correct transcription. The comparison is detailed below.

Output probability We analyze the output probability distributions of Whisper on the ASR and ST tasks of the Fleurs data (Conneau et al., 2023) on 4 language pairs: Vietnamese-English, German-English, Spanish-English and Chinese-English. Looking at Figure 2a, for ASR, most finally chosen tokens have very high probability values that are close to 1. In contrast, for ST, the probability of the finally chosen tokens spreads out much more.

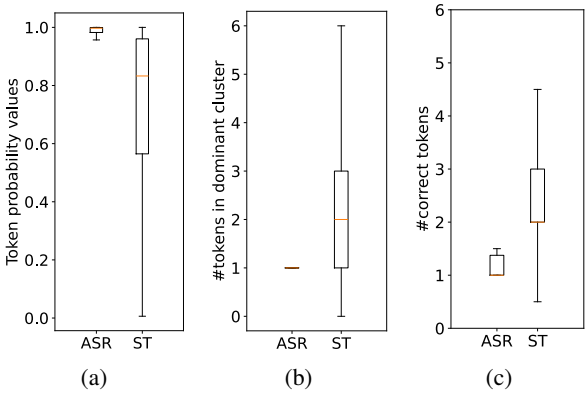

Figure 2: Model behaviours on ASR versus ST: (a) output token probability, (b) nr. dominant tokens, and (c) nr. correct tokens at every output step.

Dominant Tokens We take a closer look at the number of tokens that have notably higher probability mass in the distribution. We refer to the set of these tokens as *dominant cluster*, and the tokens themselves as *dominant tokens*. We identify them automatically using our heuristic later described in Section 4, and report on the size of the clusters in Figure 2b. Observe that clusters with sizes larger than 1 only exist for the ambiguous ST task.

Valid Output Tokens We present these top tokens to human annotators, and ask them to annotate which tokens are valid output (details in Appendix A). Looking at Figure 2c, at each output step, most of the time, there is only 1 correct output for ASR, but more than 1 for ST. This indicates that, the more spread-out probability distribution and the existence of dominant clusters with size larger than

1 are indeed due to the ambiguity of the ST task.

Underconfidence in Ambiguous Tasks Our analysis shows that, text-generation tasks with ambiguity introduce aleatoric uncertainty, i.e., uncertainty coming from the data, which differs from epistemic uncertainty, i.e., uncertainty coming from the model's incompetence. Aleatoric uncertainty makes the model appear underconfident, as the probability mass is spread over multiple valid options. We discuss this *underconfidence* phenomenon more formally with a theoretical analysis of the softmax function in Appendix B, where we show that there exists an upperbound of the probability scores assigned to every correct token at an output step, which is dependent on the number of correct tokens. This observation brings us to a simple modification to the model probability to improve its effectiveness as a quality estimator, as detailed below.

4 BOOSTEDPROB

We propose **BOOSTEDPROB**, a Quality Estimation approach which boosts the confidence of the tokens in the dominant clusters. The overall idea is that, when the output token is dominant, instead of using its own probability as the quality score, we use the total probability mass of the dominant cluster.

Finding Dominant Tokens First, we identify which tokens are in the dominant cluster given the output distribution. Previous methods designed for sampling might mistakenly account for tokens with very low probability as dominant if they happen to, e.g., be in the top-k of the probability distribution, or fall within the top-p cumulative probability mass. For sampling, this might not be a big issue, since tokens with very low probability are unlikely to be selected anyway. However, for QE, it is problematic since we would mistakenly boost the confidence of low-quality output tokens. Therefore, we propose a heuristic that looks for the dominant tokens in a stricter manner. We look for a sudden drop in the sorted probability values in order to separate dominant from non-dominant tokens.

In particular: let $X = x_1, .., x_{|X|}$ be the input sequence, and $Y = y_1, .., y_{|Y|}$ be the model output. At an output step t, let the model probability distribution over the vocabulary V be $\mathcal{P} = (p_1, p_2, \ldots, p_{|V|})$, where $p_i = \mathbb{P}(y_t = w_i \mid y_{<t}, X)$ is the probability assigned to token w_i at step t. First, we sort the probability distribution $\mathcal{P}$:

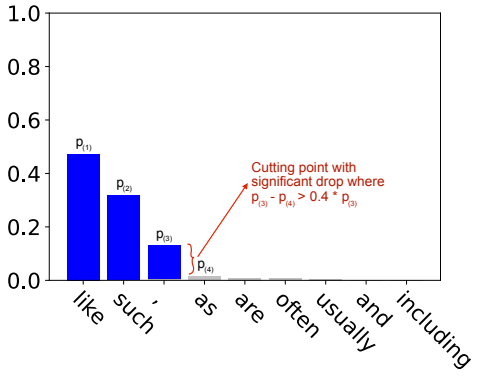

Figure 3: A dominant cluster found by our heuristic.

$$\mathcal{P}_{\text{sorted}} = (p_{(1)}, p_{(2)}, \ldots, p_{(|V|)}),$$

where $p_{(1)} \geq p_{(2)} \geq \cdots \geq p_{(|V|)}$. Then, we calculate the drops at each position, i.e., the differences between two consecutive probability values:

$$\begin{aligned}\mathcal{P}_{\text{diff}} &= \mathcal{P}_{\text{sorted}} - \text{Shift}(\mathcal{P}_{\text{sorted}}) \\ &= (p_{(1)}, p_{(2)}, \ldots, p_{(|V|-1)}) \\ &\qquad - (p_{(2)}, p_{(3)}, \ldots, p_{(|V|)})\end{aligned}$$

We then check at which positions the drops are significant. We propose a heuristic: if the drop is larger than $x\%$, then it is significant:

$$\begin{aligned}&\mathcal{P}_{\text{isSignificantDrop}} = \mathcal{P}_{\text{diff}} > \mathcal{P}_{\text{sorted}} \times x\% \\ &= (p_{(i)} - p_{(i+1)} > p_{(i)} \times x\% \;\text{ for } i = 1..|V|-1)\end{aligned}$$

Towards the distribution tail, the probabilities get close to zero, thus many drops satisfy the above condition although they are not significant drops that intuitively separate dominant from non-dominant tokens. Thus, we add another condition: the drop itself should be larger than a threshold ϵ:

$$\begin{aligned}&\mathcal{P}_{\text{isSignificantDrop}} \\ &= (\mathcal{P}_{\text{diff}} > \mathcal{P}_{\text{sorted}} \times x\%) \text{ AND } (\mathcal{P}_{\text{diff}} > \epsilon) \\ &= (p_{(i)} - p_{(i+1)} > \max(p_{(i)} \times x\%, \epsilon) \\ &\quad \text{for } i = 1..|V|-1)\end{aligned}$$

We arrived at a condition that considers both the relative value (drops larger than $x\%$) and absolute value (drops larger than ϵ), making our approach more flexible in finding the dominant tokens in different probability distributions.

The last significant drop is then the cutting point:

$$c = \max\{i \mid \mathcal{P}_{\text{isSignificantDrop\_i}} = True\}$$

where tokens with probabilities above the cutting point are dominant, and others are non-dominant. An illustration is shown in Figure 3.

Extracting Token Quality Score If the final output token is non-dominant, then we consider its own probability as the quality score. If the finally selected token is dominant, we consider the total probability mass of the whole dominant cluster as the quality score. Particularly:

$$QE(w_{(i)}) = \begin{cases} p_{(i)}, & \text{if } i > c \\ \sum_{j=1}^{c} p_{(j)}, & \text{otherwise } i \leq c \end{cases}$$

In this way, we favor the dominant tokens whose probability mass was spread amongst multiple sensible options, as described in Section 3.

Extracting sequence quality estimation The QE score for the output sequence $Y = y_1, .., y_{|Y|}$ is defined as the average of token-level QE scores:

$$QE(Y) = \sum_{t=1}^{|Y|} QE(y_t)$$

We theoretically show that BOOSTEDPROB helps tackle the *underconfidence* phenomenon discussed in Section 3 by allowing multiple correct output tokens to be assigned with a high, arbitrarily close to 1 score, which we detail in Appendix B.

5 Experimental Setup

We test BOOSTEDPROB on different tasks: Speech Translation (ST), Machine/Text Translation (MT), Summarization (Sum.), Question Answering (QA).

5.1 Data

The datasets are listed in Table 1. All datasets contain the input and ground truth output. One exception is WMT22 General (Kocmi et al., 2022), which additionally contains candidate translations of participants in the WMT22 Shared Task, along with human-annotated quality scores (0 to 100) on the **segment level**. Another exception is HJQE (Yang et al., 2023), which additionally contains model translation output from the WMT20 QE Shared Task (Specia et al., 2020) along with human-annotated quality labels (OK/BAD) on **token level**.

5.2 Models

The models used are listed in Table 2. *DeltaLM Large* is fine-tuned on 5M samples of ParaCrawl MT data, filtered by Bicleaner AI (Zaragoza-Bernabeu et al., 2022; de Gibert et al., 2024). Llama 3.3 70B is used with 4-bit quantization.

For smaller models, i.e., Whisper and DeltaLM, we generate output using beam search with beam size 4. For other models, we generate output with greedy search.

Task *	Dataset	#samples	Language
ST	Fleurs (Conneau et al., 2023)	350	vi-en, de-en, es-en, zh-en
MT	ParaCrawl (Bañón et al., 2020)	5000	en-de, zh-en
	WMT22 General (Kocmi et al., 2022)	2000	en-de, zh-en
	HJQE (Yang et al., 2023)	1000	en-de, en-zh
Sum.	XSum (Narayan et al., 2018)	3000	en
QA	GSM8k (Cobbe et al., 2021)	3000	en
	SciEx (Dinh et al., 2024a)	1120	en,de

Table 1: Data used in our experiments.

Task *	Model	Size
ST	Whisper Large V3 (Radford et al., 2023)	1550M
MT	DeltaLM Large (Ma et al., 2021)	1374M
	NLLB (Costa-Jussà et al., 2022)	3.3B
	Tower (Alves et al., 2024)	7B
Sum.	Bloomz (Muennighoff et al., 2023)	560M
+ QA	Llama 3.2 (Touvron et al., 2023)	3B
	Llama 3.3 Instruct (Touvron et al., 2023)	70B

Table 2: Models used in our experiments.

5.3 Baselines

Probability-based baselines We consider *raw model probability*, which uses the probability of the final output tokens, and *probability entropy*, which uses the entropy of the whole probability distribution. These baselines are the most comparable to our approach, as they require only the probability distributions. We use them as the main baselines throughout our experiments.

In some setups, we also consider more complex baselines, as detailed below.

Supervised QE Baseline For some translation tasks, we use a supervised QE model, WMT22 CometKiwi DA (Rei et al., 2022). The model is trained on tuples of (SRC, MT, DA), where SRC is the source sentence, MT is the MT output, and DA is the Direct Assessment scores by humans. Note that this kind of supervised QE is mostly common for translation. For other tasks like summarization

or question-answering, it is costly and not common to obtain such human-annotated quality.

Unsupervised, Ensemble-based Baselines For the word-level QE task on MT, we compare our approach with Perturbation-based QE (Dinh and Niehues, 2023), which makes minimal perturbations on the source input and measures the changes in the output as an indication of quality. For a subset of the experiments, we compare our approach with Monte Carlo sequence entropy (Malinin and Gales, 2021; Kuhn et al., 2023), which samples several output sequences and computes sequence-level entropy. These baselines are much more costly, as they require the generation of multiple outputs.

LLM self-judge We also compare our approach against a recent baseline, LLM-as-a-Judge (Zheng et al., 2023). To make this baseline more comparable to our reference-free QE, no-external-model setting, we adapt it to an LLM self-judge setup, where the model is asked to assign a quality score to its own output. This method requires an additional inference step and is not applicable to task-specific models such as Whisper and NLLB.

5.4 Hyperparameters

We loosely tune the hyperparameters, i.e., x and ϵ, on three models: Whisper, DeltaLM Large and Tower (see Appendix C). For these models, $x = 30\%$ and $\epsilon = 0.005$ are either the best or close to the best set of hyperparameters, showing that our approach is robust to hyperparameter setup. We then use these hyperparameters for all experiments.

5.5 Evaluation

On the segment level, we use Pearson correlation to measure how well QE methods correlate with the gold quality annotation. On the token level (HJQE dataset with OK/BAD labels), we use the Matthews correlation coefficient (MCC) scores (Matthews, 1975). The gold quality annotation is either automatically generated, or annotated by humans on pre-generated model output, as detailed below.

5.5.1 Automatically Generated Gold Quality

We create pseudo ground-truth quality scores to evaluate our reference-free QE methods using reference-based metrics. Reference-based metrics use human ground-truth answers in order to assign a quality score to a model output. We expect reference-based metrics to produce more reliable quality scores compared to reference-free QE methods, thus choosing them as pseudo ground-truth for evaluation.

Speech and Text Translation We use XCOMET-XL (Guerreiro et al., 2024) to generate pseudo ground-truth quality for translations. XCOMET-XL is a reference-based neural model. Dinh et al. (2024b) showed that, for MT, such reference-based neural metrics are good enough to be used as the ground truth to rank reference-free QE metrics.

Summarization and Question Answering We use BART Score (Yuan et al., 2021) as pseudo ground-truth output quality. The quality scores are calculated as the BART (Lewis et al., 2020) model probability of the output given the input text. Unlike for MT, there has not been any study showing that reference-based metrics like Bart Score are sufficient as the ground truth for reference-free QE metrics. Therefore, we additionally report on other reference-based metrics in Appendix E, including RougeL (Lin, 2004), BertScore (Zhang et al., 2019), and LLM-as-a-Judge (Zheng et al., 2023) with Qwen2.5 72B Instruct (Team, 2024).

5.5.2 Human-labeled Gold Quality

As described in Section 5.1, the WMT22 General and the HJQE datasets contain human-annotated quality labels on pre-generated output. To utilize these labels, we use the translation models of consideration to re-generate the output presented in these datasets with forced decoding, also known as reference-free Prism (Thompson and Post, 2020).

6 Results and Discussion

6.1 Overall Performance

The overall performance of BOOSTEDPROB, in comparison with the *raw probability* and *probability entropy* baselines, is shown in Table 3. BOOSTEDPROB consistently outperforms *raw probability* by a large margin (+0.194 Pearson correlation on average). *Probability entropy* appears to be a stronger baseline. This is expected since it takes into account the whole probability distribution at each output step. However, unlike BOOSTEDPROB, *probability entropy* does not consider which token was finally selected. Therefore, BOOSTEDPROB on average still has better performance than *probability entropy* (+0.065 in Pearson correlation).

The performance of BOOSTEDPROB is consistent for translation. It obtains more than 0.2 Pear-

	Model	Test Set	Language	Probability	Entropy	BOOSTEDPROB (OURS)
Speech Translation	Whisper	Fleurs	vi-en	0.112	0.379	**0.417**
		Fleurs	de-en	0.213	**0.402**	0.385
		Fleurs	es-en	0.193	0.295	**0.319**
		Fleurs	cmn-en	0.053	0.387	**0.424**
Machine Translation	DeltaLM	WMT22 General	en-de	0.165	0.169	**0.319**
		WMT22 General	zh-en	0.253	0.082	**0.688**
	NLLB	WMT22 General	en-de	0.141	0.480	**0.525**
		WMT22 General	zh-en	0.182	0.211	**0.289**
	Tower	WMT22 General	en-de	0.158	0.399	**0.414**
		WMT22 General	zh-en	0.005	0.232	**0.240**
Summarization	Bloomz 560M	XSum	en	-0.003	0.176	**0.210**
	Llama3.2 3B	XSum	en	0.002	0.201	**0.209**
	Llama3.3 70B	XSum	en	0.001	0.000	**0.004**
Question Answering (Math)	Bloomz 560M	GSM8K	en	-0.002	**0.111**	0.009
	Llama3.2 3B	GSM8K	en	-0.007	0.006	**0.111**
	Llama3.3 70B	GSM8K	en	-0.001	0.005	**0.006**
Question Answering (University Exam)	Bloomz 560M	SciEx	en,de	-0.002	0.005	**0.006**
	Llama3.2 3B	SciEx	en,de	0.002	0.228	**0.310**
	Llama3.3 70B	SciEx	en,de	0.103	**0.180**	**0.180**
Average				0.083	0.208	**0.268**

Table 3: Performance of QE methods, in Pearson correlation to gold quality, across different tasks, models, test sets.

son correlation across all settings. On the other hand, we observe cases where the two baselines fail. On Fleurs *zh-en* with Whisper and WMT22 General *zh-en* with Tower, *raw probability* has very low performance, at 0.053 and 0.005 in Pearson correlation, respectively, while BOOSTEDPROB achieves 0.424 and 0.240. On WMT22 General *zh-en* with DeltaLM, *probability entropy* obtains 0.082 score, while BOOSTEDPROB achieves 0.688. This is possibly due to BOOSTEDPROB looking at both the whole probability distribution as well as which token is selected, differing from the two baselines.

The performance on Summarization and Question Answering is more inconsistent. In some settings, all methods have very low performance, under 0.1 in Pearson correlation. This could be due to the complexity of the task, or Bart Score gold quality labels are not sufficient to rank QE methods.

We also compare BOOSTEDPROB with more advanced methods, namely Monte Carlo sequence entropy (Malinin and Gales, 2021; Kuhn et al., 2023) and LLM self-judge. As shown in Table 4, the results are mixed: our approach outperforms these baselines in certain settings. It is important to note, however, that these baselines are more costly and less flexible. Monte Carlo sequence entropy requires generating multiple output samples, while LLM self-judge requires an additional inference pass and is not applicable to task-specific models.

[1]Tower is an LLM; however, we often observe that it fails in our specific self-judge setting, i.e., fails to produce a single quality score for a given translation. This is likely because the version of Tower we used was not trained for this task.

	Model	Lang.	Monte Carlo	LLM Self judge	Boosted Prob
MT	NLLB	en-de	0.303	-	**0.525**
		zh-en	**0.337**	-	0.289
	Tower	en-de	0.302	-	**0.414**
		zh-en	**0.240**	-	**0.240**
Sum.	Bloomz 560M	en	**0.236**	0.149	0.210
	Llama3.2 3B	en	0.005	0.159	**0.209**
QA	Bloomz 560M	en	0.003	**0.019**	0.009
	Llama3.2 3B	en	0.175	**0.232**	0.111

Table 4: BOOSTEDPROB vs. Monte Carlo entropy and LLM self-judge. LLM self-judge is not applicable for task-specific models.[1]

One potential concern is whether the differences in the inference process affect the reported results. BOOSTEDPROB is designed to be applicable across inference methods (e.g., greedy decoding, beam search, top-p sampling), since it does not rely on the model always selecting the top-probability token. Furthermore, to address potential variability due to randomness, we provide example results across multiple runs with different random seeds in Appendix D.

6.2 Scoring Other Models' Output (Prism)

We evaluate BOOSTEDPROB on top of reference-free Prism: using MT models to score other translations with forced decoding. As scoring models, we use the model presented in the Prism paper, and NLLB, as it is a strong multilingual MT model, following the previous work of Zouhar et al. (2024).

We make use of the WMT22 General Shared Task data. We select the best and the worst participation systems from the shared task, by taking the

Scoring Model Scored Model	Prism Original Best MT	 Worst MT	NLLB Best MT	 Worst MT	 **Avg.**
en-de					
Probability	0.020	0.130	0.068	0.061	0.070
Entropy	0.056	0.147	0.123	0.318	0.161
BOOSTEDPROB	0.032	0.147	0.129	0.384	0.173
Supervised QE	0.202	0.453	0.202	0.453	**0.328**
zh-en					
Probability	0.205	0.285	0.153	0.085	0.182
Entropy	0.246	0.299	0.095	0.194	0.209
BOOSTEDPROB	0.251	0.317	0.153	0.231	0.238
Supervised QE	0.341	0.429	0.341	0.429	**0.385**

Table 5: Performance of QE methods with Prism, in Pearson correlation to human-labeled quality score.

average of the human-labeled quality scores on all outputs of each system. We refer to them as *Best MT* and *Worst MT*. We calculate the correlation between the QE scores to the human-labeled score.

From Table 5, we can see that BOOSTEDPROB brings improvement on top of Prism, which originally made use of the raw model probability. As a result, it shrinks the gap between Prism and the more costly supervised QE baseline.

6.3 Word-level Quality Estimation

We evaluate QE methods on annotating pre-generated translations with OK/BAD quality labels on HJQE. We again use Prism and NLLB as scoring models. We also use the original models that generated the translations in HJQE. As the QE methods provide a continuous score, we use the development split of HJQE to find the best threshold to convert the scores to the OK/BAD labels.

	en-de	en-zh	Avg.
NLLB			
Probability	0.201	0.094	0.147
Entropy	-0.042	0.007	-0.017
BOOSTEDPROB	0.204	0.123	0.164
Prism			
Probability	0.157	0.084	0.120
Entropy	-0.010	0.037	0.013
BOOSTEDPROB	0.193	0.115	0.154
Original Model			
Probability	0.143	0.176	0.159
Entropy	-0.074	-0.142	-0.108
BOOSTEDPROB	0.146	0.232	0.189
Perturbation-based QE	0.120	0.215	0.167
Supervised QE	**0.220**	**0.257**	**0.239**

Table 6: Performance of token-level QE in MCC scores.

The QE performance in MCC score is shown in Table 6. We again observe that BOOSTEDPROB achieves the best performance among the probability-based QE methods, and comes closer to the performance of the supervised QE. In this experiment, we can see that the *probability entropy* baseline fails. This is probably due to this baseline not considering the final output token. When evaluating on the sentence level, we hypothesize that the *probability entropy* would at least indicate the quality of the model prefix during autoregressive generation, thus having reasonable performance, while failing completely in this case where each token is evaluated independently.

Using the original MT model, with BOOSTEDPROB, we outperform Perturbation-based QE on the *en-zh* language pair. Note that with BOOSTEDPROB, we only need a single inference pass, unlike the Perturbation-based QE baseline.

6.4 Effect of Generative Performance

We investigate how BOOSTEDPROB works for models of different quality. We investigate this on Speech Translation, with Whisper models of varying sizes for a more controlled experiment: Whisper Tiny, Whisper Base, Whisper Small, Whisper Medium, and Whisper Large V3. We run the models on the Fleurs test set on four different language pairs as before. We report the model translation performance (in XCOMET) and the QE performance (in Pearson) alongside in Figure 4.

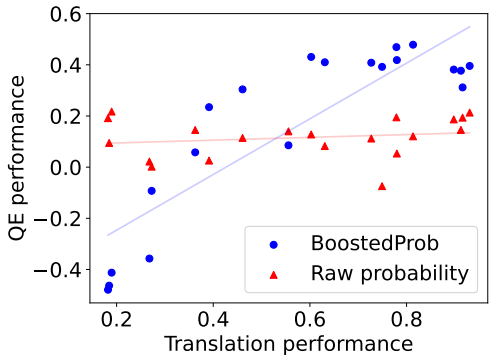

Figure 4: Model quality versus QE performance.

We see that the QE performance of the raw model probability stays consistently low. However, with BOOSTEDPROB, the QE performance improves as the quality of the model improves. This indicates that, with BOOSTEDPROB, improving a model's quality would come with improving its ability to do self-evaluation. Figure 4 also exposes a limitation of our approach, since it worsens the QE performance compared to raw probability for very weak models. The reason might be due to our approach mistakenly emphasizing the weak models' overconfidence (Appendix F). In

these cases, we recommend using a stronger model to score the output of the weak model, rather than using the probability of the weak model to score itself, like in the Prism setting (Section 6.2).

6.5 Finding Dominant Cluster

	MCC score		Best hyperparams *	
	en-de	en-zh	en-de	en-zh
Prism				
top-k	0.190	0.107	k=7	k=7
ϵ-cut	0.189	0.112	ϵ=0.01	ϵ=0.005
η-cut	0.147	0.100	ϵ=0.005	ϵ=0.01
top-p	0.119	0.071	p=0.7	p=0.95
min-p	0.156	0.085	p=0.95	p=0.90
jump-cut (ours)	**0.193**	**0.115**	x=0.3	x=0.3
			ϵ=0.005	ϵ=0.005
NLLB				
top-k	0.203	**0.123**	k=5	k=5
ϵ-cut	0.203	0.121	ϵ=0.05	ϵ=0.005
η-cut	0.180	0.105	ϵ=0.2	ϵ=0.01
top-p	0.171	0.066	p=0.7	p=0.95
min-p	0.199	0.095	p=0.7	p=0.80
jump-cut (ours)	**0.204**	**0.123**	x=0.2	x=0.3
			ϵ=0.01	ϵ=0.005

* Best hyperparameters found on the dev split.

Table 7: BOOSTEDPROB's token-level QE performance when using different methods to find dominant clusters.

We compare our method with other common methods, originally used for sampling (see Section 2.2 and 4), to find the dominant tokens for BOOSTEDPROB. We again experiment on token-level QE on HJQE. We use the HJQE development split to find the best hyperparameter for each method.

The results are shown in Table 7. We denote our method as *"jump-cut"*. Our method performs generally better than others, however, not by a large margin. Surprisingly, top-k performs comparably to our approach despite naively assuming the dominant cluster size to be fixed. This might be due to two reasons. Firstly, the HJQE dev and test sets are potentially similar, thus tuning a good k value is enough to achieve good performance. Secondly, as tokens with very low probability are unlikely to be chosen as the final output, it does not bring notable negative effect in terms of MCC score for top-k.

However, as discussed in Section 4, our method would still be a safer choice, as it would avoid mistakenly considering very low probability tokens as dominant, in the rare case that they are selected as the final output. Additionally, our method is more robust to hyperparameters: in three out of four settings in Table 7, we arrive at $x = 30\%, \epsilon = 0.005$, which are the same hyperparameter values we found before (Appendix C).

7 Conclusion

In this paper, we first perform automatic and manual analysis showing the existence of dominant clusters with sizes larger than 1 in model output probability distributions, which happens for ambiguous text-generation tasks. We show that the tokens in the dominant clusters are **underconfident**, as their probability is spread between multiple valid options. We proposed **BOOSTEDPROB** - a QE method that boosts the confidence of the dominant tokens. Since BOOSTEDPROB only utilizes the model probability distribution, it is low-cost, easy to implement, and can be applied to many model architectures. We show that BOOSTEDPROB performs notably better than model probability and probability entropy. It is also reaching close to or outperforming more costly approaches like supervised or ensemble-based QE in certain settings. With BOOSTEDPROB, improving models' quality comes with improving their self-evaluation ability.

8 Limitations

BOOSTEDPROB does not work well with very weak models, as it might emphasize weak models' overconfidence. This is mentioned in Section 6.4, and further discussed in Appendix F. In these cases, we recommend using BOOSTEDPROB with a strong model to score the output of the weak model. This is the setting of Prism, which we have reported on in Section 6.2. BOOSTEDPROB is also unlikely to have an effect for less ambiguous text generation tasks like Automatic Speech Recognition, or multiple-choice Question-Answering, since the dominant clusters with sizes larger than one are unlikely to appear (see Appendix G). One can also argue that, unsupervised QE methods are in general not very useful whenever a supervised model exists (like in Speech Translation and Machine Translation). However, in these cases, our approach still brings benefits in terms of simplicity in implementation and inference time, compared to having an extra module for supervised QE in a translation pipeline.

Acknowledgements

This work was supported by the Helmholtz Programme-oriented Funding, with project number 46.24.01, project name AI for Language Technologies. We acknowledge the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Wurttemberg and by the Federal

Ministry of Education and Research. This work also received support from the European Union's Horizon research and innovation programme under grant agreement No 101135798, project Meetween (My Personal AI Mediator for Virtual MEETtings BetWEEN People).

References

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *Preprint*, arXiv:2402.17733.

Terry Amorese, Claudia Greco, Marialucia Cuciniello, Rosa Milo, Olga Sheveleva, and Neil Glackin. 2023. Automatic speech recognition (asr) with whisper: Testing performances in different languages. In *S3C@ CHItaly*, pages 1–8.

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.

Prateek Chhikara. 2025. Mind the confidence gap: Overconfidence, calibration, and distractor effects in large language models. *arXiv preprint arXiv:2502.11028*.

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

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12621–12640, Singapore. Association for Computational Linguistics.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, Sampo Pyysalo, Stephan Oepen, and Jörg Tiedemann. 2024. A new massive multilingual dataset for high-performance language technologies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1116–1128, Torino, Italia. ELRA and ICCL.

Sourabh Deoghare, Diptesh Kanojia, Fred Blain, Tharindu Ranasinghe, and Pushpak Bhattacharyya. 2023. Quality estimation-assisted automatic post-editing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1686–1698, Singapore. Association for Computational Linguistics.

Tu Anh Dinh, Carlos Mullov, Leonard Bärmann, Zhaolin Li, Danni Liu, Simon Reiß, Jueun Lee, Nathan Lerzer, Jianfeng Gao, Fabian Peller-Konrad, Tobias Röddiger, Alexander Waibel, Tamim Asfour, Michael Beigl, Rainer Stiefelhagen, Carsten Dachsbacher, Klemens Böhm, and Jan Niehues. 2024a. SciEx: Benchmarking large language models on scientific exams with human expert grading and automatic grading. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11592–11610, Miami, Florida, USA. Association for Computational Linguistics.

Tu Anh Dinh and Jan Niehues. 2023. Perturbation-based QE: An explainable, unsupervised word-level quality estimation method for blackbox machine translation. In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 59–71, Macau SAR, China. Asia-Pacific Association for Machine Translation.

Tu Anh Dinh, Tobias Palzer, and Jan Niehues. 2024b. Quality estimation with k-nearest neighbors and automatic evaluation for model-specific quality estimation. In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 133–146, Sheffield, UK. European Association for Machine Translation (EAMT).

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. LM-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings*

of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Daniel Jurafsky and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd edition. Online manuscript released January 12, 2025.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Sergei Katkov, Antonio Liotta, and Alessandro Vietti. 2024. Benchmarking whisper under diverse audio transformations and real-time constraints. In *International Conference on Speech and Computer*, pages 82–91. Springer.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *International Conference on Learning Representations, ICLR*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Qiujia Li, David Qiu, Yu Zhang, Bo Li, Yanzhang He, Philip C Woodland, Liangliang Cao, and Trevor Strohman. 2021. Confidence estimation for attention-based sequence-to-sequence models for speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6388–6392. IEEE.

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

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of*

the Association for Computational Linguistics (ACL-04), pages 605–612, Barcelona, Spain.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *arXiv preprint arXiv:2106.13736*.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. *International Conference on Learning Representations, ICLR*.

Mouayad Masalkhi, Joshua Ong, Ethan Waisberg, Nasif Zaman, Prithul Sarker, Andrew G Lee, and Alireza Tavakkoli. 2024. A side-by-side evaluation of llama 2 by meta with chatgpt and its application in ophthalmology. *Eye*, pages 1–4.

Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Hiroki Naganuma, Ryuichiro Hataya, and Ioannis Mitliagkas. 2025. An empirical study of pre-trained model selection for out-of-distribution generalization and calibration. *Transactions on Machine Learning Research*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2024. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *arXiv preprint arXiv:2407.01082*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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.

Jan-Thorsten Peter, David Vilar, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, and Markus Freitag. 2023. There's no data like better data: Using QE metrics for MT data filtering. In *Proceedings of the Eighth Conference on Machine Translation*, pages 561–577, Singapore. 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.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yi-Lin Tuan, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Francisco Guzmán, and Lucia Specia. 2021. Quality estimation without human-labeled data. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 619–625, Online. Association for Computational Linguistics.

T Wolf. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Qianqian Xie, Qingyu Chen, Aokun Chen, Cheng Peng, Yan Hu, Fongci Lin, Xueqing Peng, Jimin Huang, Jeffrey Zhang, Vipina Keloth, et al. 2024. Me llama: Foundation large language models for medical applications. *arXiv preprint arXiv:2402.12749*.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online. Association for Computational Linguistics.

Zhen Yang, Fandong Meng, Yuanmeng Yan, and Jie Zhou. 2023. Rethinking the word-level quality estimation for machine translation from human judgement. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2012–2025, Toronto, Canada. Association for Computational Linguistics.

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

Jaume Zaragoza-Bernabeu, Gema Ramírez-Sánchez, Marta Bañón, and Sergio Ortiz Rojas. 2022. Bicleaner AI: Bicleaner goes neural. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 824–831, Marseille, France. European Language Resources Association.

Chrysoula Zerva, Frederic Blain, José G. C. De Souza, Diptesh Kanojia, Sourabh Deoghare, Nuno M. Guerreiro, Giuseppe Attanasio, Ricardo Rei, Constantin Orasan, Matteo Negri, Marco Turchi, Rajen Chatterjee, Pushpak Bhattacharyya, Markus Freitag, and André Martins. 2024. Findings of the quality estimation shared task at WMT 2024: Are LLMs closing the gap in QE? In *Proceedings of the Ninth Conference on Machine Translation*, pages 82–109, Miami, Florida, USA. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Vilém Zouhar, Shehzaad Dhuliawala, Wangchunshu Zhou, Nico Daheim, Tom Kocmi, Yuchen Eleanor Jiang, and Mrinmaya Sachan. 2023. Poor man's quality estimation: Predicting reference-based MT metrics without the reference. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1311–1325, Dubrovnik, Croatia. Association for Computational Linguistics.

Vilém Zouhar, Shuoyang Ding, Anna Currey, Tatyana Badeka, Jenyuan Wang, and Brian Thompson. 2024. Fine-tuned machine translation metrics struggle in unseen domains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 488–500, Bangkok, Thailand. Association for Computational Linguistics.

A Human Analysis of Valid Output Tokens

We perform a human analysis on the number of valid tokens at each output step. We consider the ambiguous Speech Translation (ST) task on Vietnamese-English, and the less-ambiguous Automatic Speech Recognition (ASR) task on English. We use Whisper Larger V3 output on the Fleurs test set. We identify the dominant tokens at each output step using our approach presented in Section 4, show them to the annotators, and ask the annotators to mark which tokens are the correct output. Figure 5 contains snapshots of the forms we gave to the annotators.

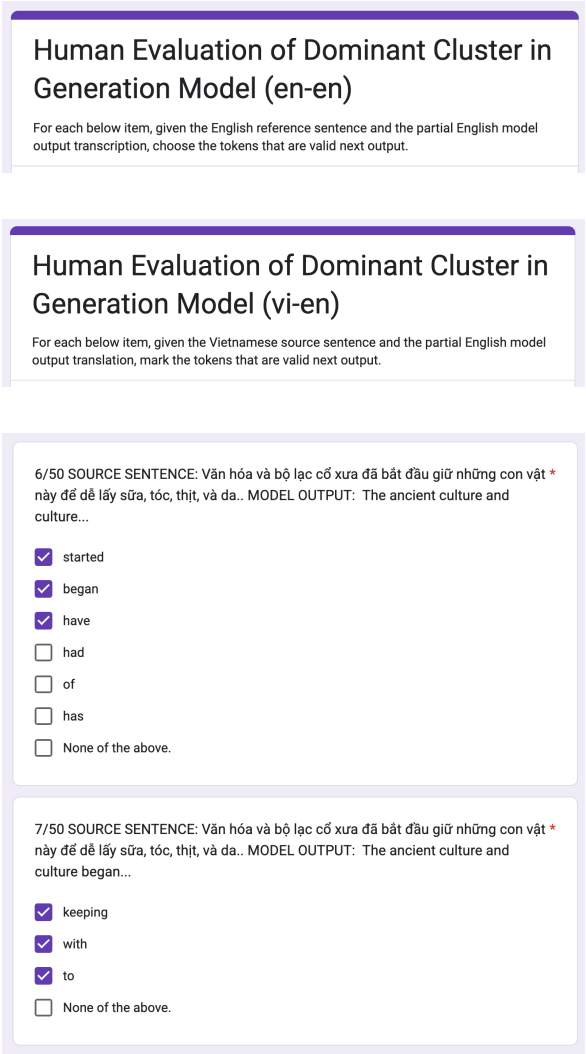

Figure 5: Snapshots of the forms provided to human annotators.

In this study, we presented 50 samples to two annotators and reported their average responses. Both annotators are students, native Vietnamese speakers with undergraduate and postgraduate education conducted in English, ensuring high proficiency in both languages. They voluntarily participated in this study without payment, and agreed to their responses being published in this paper.

The results on the number of valid tokens at each output step are already discussed in Section 3, Figure 2c. Additionally, we calculate the portion of tokens within the dominant clusters that are actually valid output. We obtained 89.02% for the ST task, and 66.86% for the ASR task. This again shows the effectiveness of our approach to identify tokens that are important within the output distributions for ambiguous tasks, and its weaker usage for less ambiguous tasks.

B Theoretical Analysis

Language models generate text by predicting the next token y_t given the previous context $y_{<t}$ and the input X, using conditional probability:

$$P(y_t|y_{<t}, X) = softmax(z_t)$$

where $z_t \in R^{|V|}$ is the logit vector at output step t over the vocabulary V.

The softmax function defines the probability of the selected token $w_i \in V$ as:

$$P(y_t = w_i|y_{<t}, X) = \frac{e^{z_{t,i}}}{\sum_{j=1}^{|V|} e^{z_{t,j}}}$$

The softmax function satisfies: $P(y_t = w_i) \geq 0$ and $\sum_{i=1}^{|V|} P(y_t = w_i) = 1$.

In natural language, multiple tokens can be the valid next output. Let:

- $C = \{w_{c1}, w_{c2}, ...\} \subset V$: the set of correct tokens at step t
- $|C| = k$: number of correct tokens.

If we want each correct token to have a minimum probability p_{min}, then:

$$\sum_{i \in C} P(y_t = w_i) \geq k \cdot p_{min}$$

However, since $\sum_{i \in V} P(y_t = w_i) = 1$, we must have:

$$k \cdot p_{min} \leq 1 \quad \Rightarrow \quad p_{min} \leq \frac{1}{k}$$

Thus, as k increases, the maximum assignable probability to each correct token decreases, leading to underconfidence.

With BOOSTEDPROB, we instead use the sum of the probability mass of C as the quality score:

$$
\begin{aligned}
&BooostedProb(y_t = w_{c1}) \\
&= BooostedProb(y_t = w_{c2}) \\
&= ... \\
&= P_C = \sum_{i \in C} P(y_t = w_i | y_{<t}, X)
\end{aligned}
$$

The only condition on P_C is that $P_C \leq 1$. Therefore, with BoostedProb, p_{min} can have a high value close to one, regardless of the size of C, thus tackling the underconfidence issue.

C Hyperparameters Tuning

C.1 Finding Dominant Tokens

We tune the hyperparameters for our approach, i.e., the value of $x\%$ that defines the relative threshold, and the value of ϵ that defines the absolute threshold that makes a reduction in probability mass significant, thus separating dominant from non-dominant tokens. The set of candidate values for $x\%$ is $\{0.2, 0.3, 0.4, 0.5, 0.6\}$. The set of candidate values for ϵ is $\{0.005, 0.01, 0.1\}$.

We perform hyperparameter tuning on the development splits of the datasets: 5,000 samples from ParaCrawl for each of the language pairs *en-de* and *zh-en* for Machine Translation, and the pre-defined development split of Fleurs on 4 language pairs *de-en, es-en, vi-en, zh-en* for Speech Translation. We tune for three models: Whisper Large V3, DeltaLM and Tower, take the average over all language pairs, and report the results in Figure 6, Figure 7 and Figure 8, respectively.

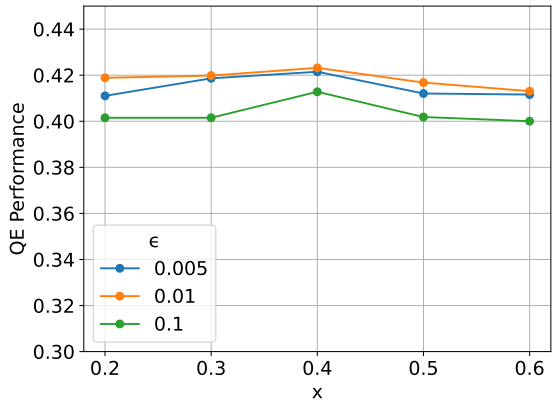

Figure 6: Hyperparameter tuning on Whisper Large V3.

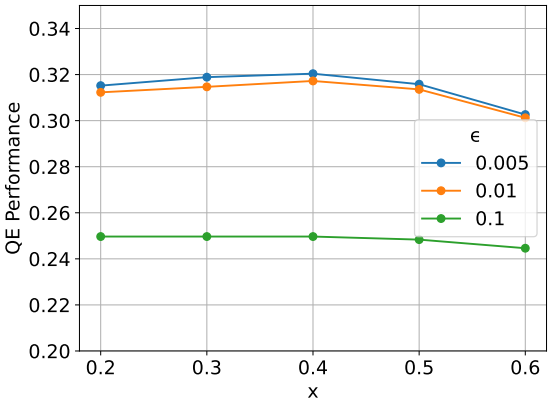

Figure 7: Hyperparameter tuning on DeltaLM Large.

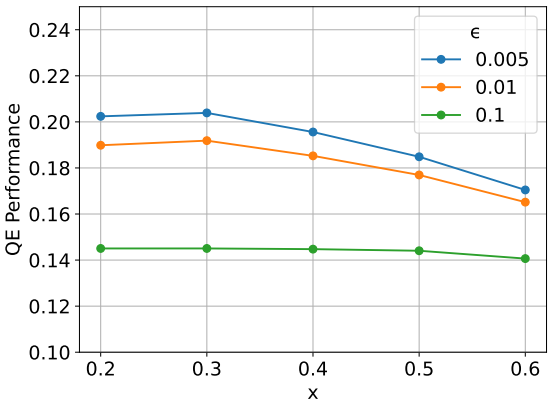

Figure 8: Hyperparameter tuning on Tower.

As can be seen from the plots, for all models, the hyperparameter set with $x = 0.3$ and $\epsilon = 0.005$ gives the best or close to the best performance. This shows that our approach is rather robust to hyperparameters. As a result, we use these values for all of our experiments in the paper.

C.2 Sequence Aggregation of Token Scores

We also experiment with different ways to aggregate token-level scores to sequence-level scores with our approach. We try taking the mean, the median, and the minimum of the token-level scores. We also try counting the number of output tokens in the sequence that are within the dominant clusters, rather than using the probability mass of the clusters.

The results are shown in Table 8. As can be seen, using the probability mass is better than simply counting the number of dominant tokens. Taking the mean of the token scores is better than taking the median, potentially due to the median ignoring catastrophic token-level errors. Taking the

minimum of the token scores ended up giving better performance than taking the mean, indicating that the quality of the lowest-score token might be sufficient to represent the quality of the whole sequence. However, we stick with using the mean as the token-to-sequence score aggregation method in our experiments, in order to have a more global view of the whole sequence quality.

	Whisper	DeltaLM	Tower	**Avg.**
Mean	0.419	0.319	0.204	0.314
Median	0.272	0.253	0.147	0.224
Min	0.439	0.436	0.236	**0.370**
Nr. Dominant	0.107	0.315	0.216	0.213

Table 8: Different token-to-sequence scrore aggregation methods.

D Randomness in Inference

To account for randomness in the inference process, it is ideal to repeat each experiment with different random seeds and report the variance. However, this approach is computationally expensive. In our work, we instead prioritize running a broad set of experiments across models and tasks, rather than repeating each experiment multiple times. For illustration, we conduct an additional study with multiple seeds in one setting: the NLLB model on a translation task. Specifically, we apply top-p sampling with $p = 0.5$ and repeat the experiment using five different seeds (integers from 0 to 4). The results, which incorporate performance variance, are presented in Table 9 and demonstrate that our approach consistently outperforms the baselines.

Lang.	Probability	Entropy	BOOSTEDPROB
en-de	0.142 ± 0.002	0.449 ± 0.030	**0.481** ± 0.039
zh-en	0.184 ± 0.004	0.226 ± 0.010	**0.312** ± 0.010

Table 9: Performance of QE methods in Pearson correlation across runs with different random seed on NLLB translation.

E Correlation with ROUGE-L and BertScore

Since relying on a single reference-based metric - specifically BARTScore in our main experiments - may not provide a sufficiently robust ground truth for evaluating reference-free Quality Estimation approaches, we additionally report results using ROUGE-L (Lin and Och, 2004) and BERTScore (Zhang et al., 2019). We also repeat the results on Bart Score, which we presented in Section 6.1, for a more complete overview.

The results are shown in Table 10. With all three ground truth metrics, we generally observe similar patterns: our BOOSTEDPROB approach performs better than the raw probability and the probability entropy. One exception is on the Question Answering task evaluated with Bert Score as ground truth, where all QE approaches give a negative correlation most of the time. This is possible due to the GSM8k test set is about solving math problems. Bert Score might not be a suitable metric, since it compares the contextual embeddings of the model output to the reference, which might not emphasize critical errors with wrong number output in math problems, since numbers might be close to each other in the embedding space.

F Negative Effect on Very Weak Models

As mentioned in Section 6.4, BOOSTEDPROB improves QE performance of output probability for stronger models, but worsens it for weak models. This is somewhat expected, since the motivation of BOOSTEDPROB is to improve cases when the model is underconfident. It does not consider the cases when a low-quality model is overconfident and constantly assigns high probability values to the wrong token. To test whether this is truly the cause, we manually look at some output by the worst-performing model, Whisper Tiny, on Chinese-to-English test data. One example is as follows:

Source: "有了它，我们才有了火车、汽车和许多其他交通工具"

Reference: "*It has brought us the train, the car, and many other transportation devices.*"

Model output: "*There we have it.*"

Observe that the model exhibits signs of hallucination, as the output is quite irrelevant to the input sentence and the ground-truth reference. However, when we look at the probability distributions of the output tokens, they do form dominant clusters. For example, at the third output step after "*There we ...*", the dominant next tokens assigned by the model are "*are*", "*have*" and "*go*", as shown in Figure 9. These tokens seem to be hallucinated: they are common words that might come after "*There we ...*", but are quite irrelevant to the input sentence. In cases like this, by favoring the dominant tokens, our approach emphasizes the models' overconfidence, thus leading to bad quality estimation

Ground Truth	Task	Model	Test Set	Language	Probability	Entropy	BoostedProb
Bart Score	Summarization	Bloomz 560M	XSum	en	-0.003	0.176	**0.210**
		Llama3.2 3B	XSum	en	0.002	0.201	**0.209**
		Llama3.3 70B	XSum	en	0.001	0.000	**0.004**
	Question Answering	Bloomz 560M	GSM8K	en	-0.002	**0.111**	0.009
	(Math)	Llama3.2 3B	GSM8K	en	-0.007	0.006	**0.111**
		Llama3.3 70B	GSM8K	en	-0.001	0.005	**0.006**
	Question Answering	Bloomz 560M	SciEx	en,de	-0.002	0.005	**0.006**
	(University Exam)	Llama3.2 3B	SciEx	en,de	0.002	0.228	**0.310**
		Llama3.3 70B	SciEx	en,de	0.103	**0.180**	**0.180**
RougeL	Summarization	Bloomz 560M	XSum	en	0.048	0.176	**0.216**
		Llama3.2 3B	XSum	en	-0.207	**0.632**	0.619
		Llama3.3 70B	XSum	en	0.061	0.001	**0.061**
	Question Answering	Bloomz 560M	GSM8K	en	0.049	0.107	**0.125**
	(Math)	Llama3.2 3B	GSM8K	en	-0.169	-0.054	**0.079**
		Llama3.3 70B	GSM8K	en	0.148	**0.294**	0.227
	Question Answering	Bloomz 560M	SciEx	en,de	-0.273	0.473	**0.497**
	(University Exam)	Llama3.2 3B	SciEx	en,de	0.102	0.138	**0.141**
		Llama3.3 70B	SciEx	en,de	0.182	0.204	**0.223**
Bert Score	Summarization	Bloomz 560M	XSum	en	-0.083	0.023	**0.111**
		Llama3.2 3B	XSum	en	-0.121	0.091	**0.099**
		Llama3.3 70B	XSum	en	0.070	-0.024	**0.162**
	Question Answering	Bloomz 560M	GSM8K	en	-0.022	-0.022	-0.038
	(Math)	Llama3.2 3B	GSM8K	en	-0.121	-0.257	-0.126
		Llama3.3 70B	GSM8K	en	**0.121**	-0.361	-0.257
	Question Answering	Bloomz 560M	SciEx	en,de	0.420	0.411	**0.434**
	(University Exam)	Llama3.2 3B	SciEx	en,de	0.009	0.007	**0.015**
		Llama3.3 70B	SciEx	en,de	0.005	0.164	**0.380**
LLM-as-a-Judge	Summarization	Bloomz 560M	XSum	en	0.014	0.265	**0.287**
		Llama3.2 3B	XSum	en	-0.039	0.000	**0.156**
		Llama3.3 70B	XSum	en	0.018	0.130	**0.184**
	Question Answering	Bloomz 560M	GSM8K	en	0.002	0.023	-0.013
	(Math)	Llama3.2 3B	GSM8K	en	0.015	-0.204	**0.128**
		Llama3.3 70B	GSM8K	en	0.024	-0.035	**0.053**
	Question Answering	Bloomz 560M	SciEx	en,de	0.539	0.646	**0.654**
	(University Exam)	Llama3.2 3B	SciEx	en,de	-0.030	-0.123	-0.008
		Llama3.3 70B	SciEx	en,de	0.076	0.229	**0.305**

Table 10: Correlation of different quality scores to different ground truth metrics: Bart Score, RougeL, BertScore, and LLM-as-a-Judge with Qwen2.5 72B.

performance.

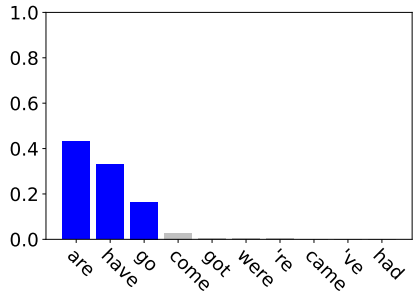

Figure 9: Example of Whisper Tiny's hallucinated probability distribution at an output step.

G Effect on Less Ambiguous Tasks

We expect that BOOSTEDPROB is unlikely to have major differences from raw model probability on less ambiguous tasks, such as Automatic Speech Recognition (ASR). Recall that for ASR, the size of the dominant clusters is usually 1 (see Section 3, Figure 2b). Therefore, using the total probability mass of the dominant cluster in BOOSTEDPROB would be the same as using the raw model probability of the single dominant token.

We confirm this hypothesis by applying BOOSTEDPROB on the ASR task with Whisper Large V3 and the Fleurs test set, as shown in Table 11. The QE performance here is the Pearson correlation of the QE scores with the Word Error Rate of the transcription output. As can be seen, the QE performance of BOOSTEDPROB is similar to that of the raw model probability.

	en	de	es	zh
Probability	0.363	0.346	0.375	0.375
BOOSTEDPROB	0.364	0.378	0.383	0.396

Table 11: BOOSTEDPROB versus raw probability on ASR tasks.

We can conclude that BOOSTEDPROB gives the same or better QE performance than the raw model probability, depending on the magnitude of ambiguity of the task at hand.

H Discussion on Overall QE correlations

As can be seen in Section 6.1 and Section 6.2, our method gives around 0.3 points on average, and 0.688 at max in Pearson correlation with the gold quality score. One can raise the question: Can this be interpreted as correlated at all? Does this mean that the proposed approach (as well as the baselines) offers very limited practical use?

Note that Pearson correlation ranges between -1 and 1. An example plot of our BoostedProb approach with NLLB on Prism, in correlation with human scores, on the WMT 22 *en-de* data is shown in Figure 10. With Pearson correlation of 0.384, we can already see the positive trend from the plot.

As can be seen in Section 6.2, even the supervised Quality Estimation model obtained around 0.4 correlation. As a broader pointer, we can consider the results of a recent public shared task on Quality Estimation for Machine Translation at WMT 2024 (Zerva et al., 2024). In their findings, page 103, appendix B, Table 8, the Pearson correlations of the participating QE systems are also around 0.4, including the SOTA system by Unbabel.

Even with this current progress of the field, Quality Estimation has already shown to be useful in many applications, e.g., guiding the decoding process to generate better translation (Fernandes et al., 2022), supporting post-editing (Deoghare et al., 2023), or to filter out synthetically created bilingual data (Peter et al., 2023).

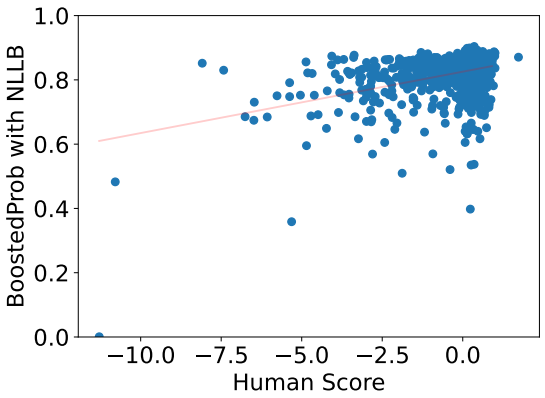

Figure 10: Illustration of the positive correlation between our method on top of Prism with NLLB and the human score, on the WMT22 General data, *en-de*.

I Tools and Hardwares

The Speech Translation experiments are conducted using Huggingface (Wolf, 2019). The Text Translation experiments are conducted using Fairseq (Ott et al., 2019). The Summarization and Question Answering experiments are conducted using LM-Polygraph (Fadeeva et al., 2023). For all experiments, we use A100 GPUs with 40GB of memory.

J License For Artifacts

The license for artifacts used in our paper is as follows:

- Fleurs dataset (Conneau et al., 2023): CC BY 4.0
- ParaCrawl dataset (Bañón et al., 2020): Creative Commons CC0
- WMT22 General dataset (Kocmi et al., 2022): Apache License 2.0
- XSum dataset (Narayan et al., 2018): MIT License
- GSM8k dataset (Cobbe et al., 2021): MIT License
- Whisper models (Radford et al., 2023): Apache License 2.0
- DeltaLM model (Ma et al., 2021): MIT License
- NLLB model (Costa-Jussà et al., 2022): CC BY NC 4.0
- Tower model (Alves et al., 2024): CC BY NC 4.0
- Bloomz model (Muennighoff et al., 2023): The BigScience RAIL License
- Llama 3.2 models (Touvron et al., 2023): Llama 3.2 Community License Agreement
- Llama 3.3 models (Touvron et al., 2023): Llama 3.3 Community License Agreement