Are Generative Models Underconfident? An Embarrassingly Simple Quality Estimation Approach

Anonymous ACL submission

Abstract

Quality Estimation (QE) is estimating the quality of model output when the ground truth reference is not available. Looking at model uncertainty from its own output probabilities is the most trivial and low-effort way to estimate the output quality. However, for generative model, output probabilities might not be the best quality estimator. At an output step, there can be multiple correct options, making the probability distribution spread out more. Thus, lower token probability does not necessarily mean lower output quality. In other words, the model can be considered underconfident. In this paper, we propose a QE approach called ***Dominant Mass Probability*** (DMP), that boosts the model confidence in cases where there are multiple viable output options. We show that, with no increase in complexity, DMP is notably better than sequence probability when estimating the quality of different models (Whisper, Llama, etc.) on different tasks (translation, summarization, etc.). Compared to sequence probability, DMP achieves on average +0.208 improvement in Pearson correlation to ground-truth quality.

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 are not perfect, as they would 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 in the research community as Quality Estimation.

Particularly, Quality Estimation (QE) is the task of providing quality scores on model output when the ground truth is not available. The most straightforward way is to infer the output quality from the model uncertainty by looking at model's output probability. However, for free-form text generation tasks, such as translation or summarization, model probability might not be the best estimator. For these tasks, there can be multiple correct outputs for a single input sequence. This leads to models being **underconfident**: lower probability does not necessarily indicate lower quality output, but could mean that the probability distribution is spread out on multiple correct options.

In this paper, we propose a simple QE approach called ***Dominant Mass Probability*** (DMP), which only utilizes the model output probability distribution. Without any added complexity, DMP tackles the underconfident phenomena mentioned above by boosting the confidence scores in the cases where there are multiple tokens with dominating probability values in the model output distribution. In particular, our contributions are as follows:

1. We perform analysis showing that there indeed exist clusters of dominant tokens in the model output distribution that lead to underconfidence for free-form text generation tasks.

2. We propose a Quality Estimation approach called ***Dominant Mass Probability*** (DMP) [1] to tackle the underconfidence phenomena. DMP is easy to implement and does not add any complexity overhead compared to using raw model output probabilities.

3. We show that DMP is notably better as a quality estimator than the raw model probabilities across different tasks and different models, with an average increase of +0.208 in Pearson correlation to the ground truth quality.

[1] Code submitted as zip file.

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 way of utilizing the model output probability for quality estimation is to calculate the entropy of the whole probability distribution (Fomicheva et al., 2020). However, probability entropy does not take into account which option is selected in the end. These methods utilizing model probabilities are generally low-effort, with the only drawback that output probability might not always be accessible for API-only models. Therefore, probability-based QE has been successfully employed in many use cases. For example, in dialog systems, the model probability of speech recognition output is used to decide whether to ask the user to repeat (Jurafsky and Martin, 2025). In early exiting models, the probability entropy is used to decide at which layer the model can stop the forward pass and output the final prediction (Teerapittayanon et al., 2016; Xin et al., 2020).

Other lines of Quality Estimation approaches are usually more costly. They either require more inference runs, such as ensemble-based approaches (Kuhn et al., 2023; Malinin and Gales, 2021) and self-validation approaches (Kadavath et al., 2022); or require access to the model training data to detect out-of-distribution instances during inference (Lee et al., 2018; Ren et al., 2023); or requires an external model to measure the output quality (Rei et al., 2022; Cohen et al., 2023).

One outstanding case of using external module for quality measure is supervised Quality Estimation models for the task of text translation. Unlike other text generation tasks, for machine translation, there exists abundant data of (source, model translation, human-labeled scores) tuples, which enable training supervised models that output quality scores. Quality Estimation has been widely adopted in the field of machine translation, and is even getting close to the performance of translation metrics that use reference ground-truth translation (Freitag et al., 2022).

2.2 Dominant Tokens

Previous works have taken into account that there can be multiple dominant tokens in the probability distribution at an output step. However, they mostly focus on the case of sampling, rather than for quality estimation. They try finding the set of dominant tokens to sample from during generation in order to maintain high quality but also have diversity in the output. 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 hidden assumption is that, the top k tokens with the highest probability are the most important ones. For top-p, the most important tokens are ones 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 $\sqrt{\eta} * exp(-entropy(\mathbb{P}))$, where $\mathbb{P}$ is output probability distribution. For min-p, the most important tokens have probabilities that is larger than the top-1 probability multiplied by p.

3 Method

3.1 Problem definition

X-to-one vs. X-to-many Our Quality Estimation method is inspired by the difference in model behavior between X-to-one and X-to-many tasks. An example of X-to-one task is Automatic Speech Recognition (ASR), where for each input audio, there is only one correct transcription. Here the models would assign most probability mass to one token that it deems correct at each output step.

In contrast, we have X-to-many tasks such as Speech Translation (ST), where for each input sentence, multiple translations can be correct. This introduces aleatoric uncertainty, i.e., uncertainty coming from the data, but not from model's incompetency. This aleatoric uncertainty makes the models appear underconfident, as they need to spread out the probability mass over multiple correct options.

Illustrative example Consider the example in Figure 1, where Whisper Large V3 translate a Vietnamese audio sentence to English. The first two cases (correct translation to "*elephants*" and wrong translation to "*raccoons*") are intuitive: higher probabilities indicate better output quality. However, in the third case, most probability mass are spread between three options: the comma ",", "*like*" and "*such as*", which are all reasonable next word. The probability values here are lower, but do not indicate low output quality.

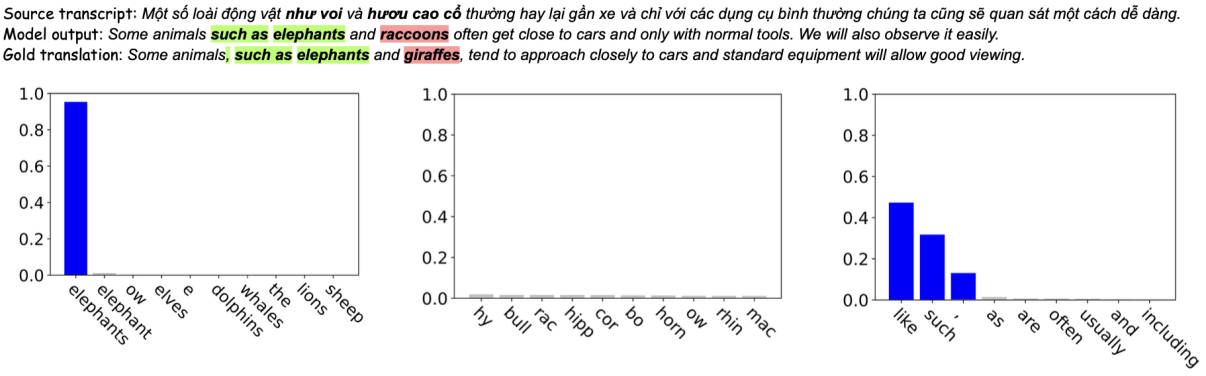
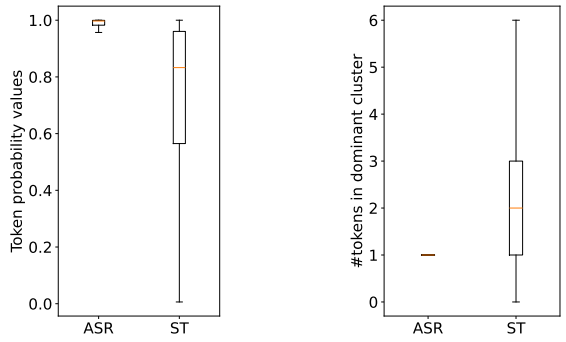


Figure 1: Example of Whisper Large V3's probability distributions when translating from Vietnamese audio to English text. In the first case, the model gives high probability to the correct translation of "*elephants*". In the second case, the model gives low probability to all tokens in the probability, and ends up outputting the wrong translation ("*raccoons*" instead of "*giraffes*"). In the last case, the probability of the tokens is lower due to probability mass being spread out between multiple correct options (the comma ",", "*like*" and "*such as*" are all reasonable next word), and ***do not indicate lower quality***.

Dominant tokens We refer to the set of tokens with the most probability mass at an output step as *dominant cluster*, and the tokens themselves as *dominant tokens*. We propose a heuristic to find these dominant clusters later in Section 3.2. We find that, dominant clusters with sizes larger than 1 only exist for X-to-many tasks. We gather the statistics from all output steps of Whisper Large V3 on the ASR and ST task of the Fleurs test set (Conneau et al., 2023), and report them in Figure 2. For the ASR task, most finally chosen tokens have very high probability values that are close to 1. On the other hand, for the ST task, the probability of the finally chosen tokens spread out much more. We can also see this from Figure 2b, where for the ASR task, most dominant clusters only contain one element. For the ST task, the number of tokens in the dominant clusters is often more than one.



(a) Probabilities of final output token at every step.

(b) #tokens in the dominant cluster at every step.

Figure 2: Comparision between the x-to-one ASR task and the x-to-many ST task.

Motivation for QE Given the above analysis, we confirm that the existence of dominant clusters with sizes larger than 1 is due to data uncertainty (aleatoric uncertainty), and not from model's uncertainty (epistemic uncertainty). The dominant tokens in these clusters would have lower probability that do not correctly reflect their quality as an output. Therefore, we propose a quality estimation approach, called ***Dominant Mass Probability*** (DMP), that favors these dominant tokens.

3.2 Quality Estimation with Dominant Mass Probability

Finding dominant token As the first step, we need to identify which tokens are in the dominant cluster given the output distribution. We propose a heuristic approach that looks for a sudden drop in the sorted probability values which separate dominant tokens 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 sequence. 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 $p_i = \mathbb{P}(y_t = w_i \mid y_{<t}, X)$ represents the probability assigned by the model to token w_i at output step t. First, we sort the values in the probability distribution $\mathcal{P}$ and obtain:

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

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

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

We then check at which positions the drops are significant. We propose a heuristic for the check: if the drop is larger than $x\%$ of the probability value (empirically, we find 30% and 40% are suitable values), then it is significant:

$$\mathcal{P}_{\text{isSignificantDrop}}$$
$$= \mathcal{P}_{\text{diff}} > \mathcal{P}_{\text{sorted}} * x\%$$
$$= (p_{(i)} - p_{(i+1)} > p_{(i)} * x\% \text{ for } i = 1..|V|-1)$$

Towards the tail of the distribution, 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. Therefore, we add another condition for the drop to be significant: the drop itself should be larger than a threshold ϵ (empirically, we find 0.01 and 0.1 are suitable values):

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

We then choose the last significant drop as the cutting point c:

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

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

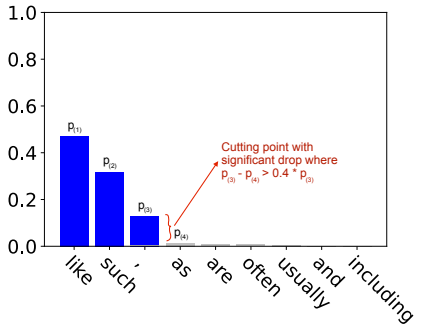


Figure 3: A dominant cluster found by our heuristic.

Extracting token quality estimation Once we have found the dominant tokens, we extract quality estimation scores. If the finally selected output token is non-dominant, then we consider its probability to be the quality score as usual. If the finally selected token is dominant, we consider the total probability mass of the whole dominant cluster as the quality score. That is, we take the sum of the probabilities of all dominant tokens 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 3.1.

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:

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

4 Experimental Setup

We test our Quality Estimation method on four different tasks: Speech Translation, Text Translation, Summarization and Question Answering.

4.1 Data

The datasets used in our experiments are listed in Table 1. All datasets contain the input and ground truth of the corresponding task. One exception is WMT22 General (Kocmi et al., 2022), which additionally contains translation output of participating systems from the WMT22 Shared Task, along with human-annotated quality score ranging from 0 to 100 on the **segment level**. Another exception is HJQE (Yang et al., 2023), which additionally contains model translation output from the WMT20 Quality Estimation Shared Task (Specia et al., 2020) along with human-annotated quality labels (OK/BAD) on the **token level**.

4.2 Models

The models used in our experiments are listed in Table 2. *Scratch* is a model trained from scratch on 5M samples from the ParaCrawl dataset, filtered by Bicleaner AI (Zaragoza-Bernabeu et al., 2022;

Task	Dataset	#samples	Language
Speech Translation	Fleurs (Conneau et al., 2023)	350	vi-en, de-en, es-en, cmn-en
Text Translation	ParaCrawl (Bañón et al., 2020)	5000	en-de
	WMT22 General (Kocmi et al., 2022)	2000	en-de
	HJQE (Yang et al., 2023)	1000	en-de
Summarization	XSum (Narayan et al., 2018)	3000	en
Question Answering	GSM8k (Cobbe et al., 2021)	3000	en

Table 1: Data used in our experiments.

Task	Model	#parameters
Speech Translation	Whisper Large V3 (Radford et al., 2023)	1550M
Text Translation	Scratch *	62M
	DeltaLM Large (Ma et al., 2021)	1374M
Summarization	Bloomz (Muennighoff et al., 2023)	560M
+ Question Answering	Llama 3.2 (Touvron et al., 2023)	3B
	Llama 3.3 Instruct (Touvron et al., 2023)	70B

Table 2: Models used in our experiments. *: *Scratch* is a transformer model trained on 5M ParaCrawl samples.

de Gibert et al., 2024). *DeltaLM Large* is fine-tuned on the Machine Translation task on the same ParaCrawl data. The Llama 3.3 70B model is used with 4-bit quantization.

4.3 Baselines

Probability-based baselines We consider 2 baselines: *sequence probability* and *mean token entropy*. *Sequence probability* is the product of token probabilities in an output sequence. *Mean token entropy* is the average entropy of all tokens in an output sequence. These two baselines are the most comparable to our approach, as they require only the probability distribution of output tokens.

Supervised Quality Estimation baseline For some translation tasks, we use a supervised Quality Estimation (QE) model, WMT22 CometKiwi DA (Rei et al., 2022). The model is trained on tuples of (SRC, MT, DA), where SRC is the input source sentence, MT is the machine translation output sentence, and DA is the Direct Assessment scores on the given by human annotators. DA scores range from 0 to 100, where 0 is assigned to the worst translation and 100 is assigned to the best translation. Note that this kind of supervised QE model trained on human-labeled quality annotations is mostly common for translation tasks. For other tasks such as summarization or question-answering, it would be more costly to obtain such human-annotated quality data. We regard this approach as an upper baseline for our approach.

4.4 Hyperparameters

We choose the hyperparameters for our approach, i.e., the values for x and ϵ, by tuning on the development splits of the datasets. We use 5000 samples from ParaCrawl for the Text Translation task, and use the development split of Fleurs for the Speech Translation task. We arrive at $x = 30\%$ and $\epsilon = 0.1$ for Text Translation tasks, and $x = 40\%$ and $\epsilon = 0.01$ for Speech Translation tasks. We observe that these values are close to each other and make little changes to the final QE performance. Therefore, we apply them directly on the remaining tasks (Summarization and Question Answering), with $x = 30\%$ and $\epsilon = 0.01$.

4.5 Evaluation

On the segment level, we use Pearson correlation to measure how well the scores from the quality estimation methods correlate with the gold quality annotation. The higher the correlation, the better the quality estimation methods perform. 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.

4.5.1 Automatically Generated Gold Quality

Speech and Text Translation We use XCOMET-XL (Guerreiro et al., 2024) as the gold quality of translation output. XCOMET-XL is a neural model trained to predict the quality of translations given the source, model translation and ground truth translation. (Dinh et al., 2024) showed that, for machine translation, such reference-based neural metrics are good enough to be used as gold quality annotation to rank reference-free QE metrics.

	Model	Test Set	Language	Probability	Entropy	DMP (Ours)
Speech Translation	Whisper	Fleurs	vi-en	0.112	0.379	**0.408**
		Fleurs	de-en	0.213	0.402	0.396
		Fleurs	es-en	0.193	0.295	**0.312**
		Fleurs	cmn-en	0.053	0.387	**0.418**
Machine Translation	Scratch	ParaCrawl	en-de	0.155	0.070	**0.221**
		WMT22 General	en-de	0.197	0.147	**0.370**
	DeltaLM	ParaCrawl	en-de	0.131	0.053	**0.386**
		WMT22 General	en-de	0.165	0.169	**0.297**
Summarization	Bloomz 560M	XSum	en	0.111	0.189	**0.215**
	Llama3.2 3B	XSum	en	0.139	**0.236**	0.227
	Llama3.3 70B	XSum	en	-0.006	0.002	0.004
Question Answering	Bloomz 560M	GSM8K	en	-0.007	0.107	**0.142**
	Llama3.2 3B	GSM8K	en	0.006	0.295	**0.366**
	Llama3.3 70B	GSM8K	en	-0.267	**0.377**	0.341
Average				0.085	0.222	**0.293**

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

Summarization and Question Answering We use BART Score (Yuan et al., 2021) to annotate the quality of each output summary. The quality scores here are calculated as the mean token log probability from BART (Lewis et al., 2019) of the summary output given the original text.

4.5.2 Human-labeled Gold Quality

As described in Section 4.1, the WMT22 General and the HJQE datasets contain human-annotated quality labels on pre-generated model output. In order to utilize these labels, we use the translation models of consideration ("Scratch" and *DeltaLM Large*) to re-generate the output from the other models from the dataset with forced decoding. In this way, we avoid the biases from using an external model (XCOMET-XL) to create gold quality score.

5 Results and Discussion

5.1 Overall Performance

The overall performance of our approach, DMP, in comparison with the *sequence probability* and *mean token entropy* baselines, is shown in Table 3. DMP consistently outperforms the *sequence probability* baseline by a large margin (+0.208 Pearson correlation on average). The *mean token entropy* appears to be a stronger baseline. This is expected since this method takes into account the whole probability distribution at each output step. However, it does not take into account which token was finally selected. Therefore, our approach on average still has better performance than *mean token entropy* (+0.071 in Pearson correlation).

The performance of our approach, DMP, is more consistent on translation tasks. It obtains > 0.2 Pearson correlation across all settings. On the other hand, we observe cases where the two baselines fail. On the ParaCrawl test set, *mean token entropy* obtains 0.070 and 0.053 Pearson correlation with the gold quality scores on the *Scratch* and *DeltaLM* model, respectively, while DMP achieves 0.221 and 0.388. On the Fleurs test set, Chinese-English translation, the *sequence probability* baseline obtained 0.053 Pearson correlation, as opposed to our approach with 0.418. This is possibly due to our approach both looking at the whole probability distribution as well as taking into account which token is selected in the end.

The performance on the Summarization and Question Answering tasks are more inconsistent. For Summarization with Llama3.3 70B, all three methods fail. For Question Answering with Llama3.3 70B, the *sequence probability* has negative Pearson correlation. This either could be due to the complexity of the task, or that using automatically created gold quality labels from BartScore is not sufficient to rank quality estimation methods.

5.2 Scoring Other Models' Output

In this experiment, we focus on evaluating the QE approaches when being used on one model to evaluate output created by other models for the Text Translation task. As described in Section 4.5.2, we use our model of consideration, i.e., *Scratch* and *DeltaLM Large*, to generate output from the other models from the dataset with forced decoding.

5.2.1 Sentence-level Quality Estimation

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 average of the human-labeled quality scores on all output sentences of each system. We refer to them as *Best*

	Best MT	Worst MT	Average
Scratch			
Probability	0.071	0.054	0.063
Entropy	0.147	0.240	0.194
Dominant	0.156	0.267	0.212
DeltaLM			
Probability	0.070	0.064	0.067
Entropy	0.161	0.308	0.235
Dominant	0.178	0.338	0.258
Supervised QE	0.202	0.453	0.328

Table 4: Performance of quality estimation methods, in Pearson correlation to human-labeled quality score

MT and *Worst MT*. We calculate the correlation between the scores from the QE approaches to the human-labeled quality score.

The results are shown in Table 4. The *sequence probability* baseline does not work in this setting, obtaining less than 0.1 Pearson correlation to the human-labeled quality scores on all settings. The *mean token entropy* baseline performs better, at 0.194 using the *Scratch* model and 0.235 using the *DeltaLM Large* model. Among the probability-based approaches, our approach has the best performance, at 0.212 using the *Scratch* model and 0.258 using the *DeltaLM Large* model. It still lags behind the supervised QE baseline by around 0.1. However, this gap is not as large as expected, given that the supervised QE baseline is more complex, in terms of both computation and training data.

5.2.2 Word-level Quality Estimation

We evaluate the performance of the QE methods on annotating pre-created output with OK/BAD quality labels on the HJQE dataset. As the probability-based quality estimation methods provide a continuous score for each token, we use the development split of HJQE to find the best threshold to convert the scores to the OK/BAD binary labels, and apply the threshold on the test set.

	HJQE dev	HJQE test
Scratch		
Probability	0.169	0.110
Entropy	0.001	-0.005
Dominant	0.197	0.134
DeltaLM		
Probability	0.234	0.138
Entropy	-0.009	0.001
Dominant	**0.280**	0.156
Supervised QE	0.240	**0.165**

Table 5: Performance of quality estimation methods on the token level, in MCC scores compared to the gold human labeled quality.

The QE performance in MCC score is shown in Table 5. We again observe that DMP achieves the best performance amongst the probability-based quality estimation methods, and closely approaches the performance of the supervised QE model. In this experiment, we can see that the *mean token entropy* baseline fails. This is probably due to the negative effect of this baseline not taking into account the final output token. When evaluating on the sentence level, we hypothesize that the *mean token 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.

5.3 Effect of Generative Performance

We investigate whether our QE methods work for models of different quality. We focus on the case of Speech Translation, where we investigate 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 same Fleurs test set on four different language pairs as before. We report the model translation performance and the quality estimation performance alongside each other in Figure 4. The model performance is calculated as the average XCOMET score over all translation segments. The quality estimation performance is calculated as the Pearson performance to the segment-level XCOMET scores, similar to before.

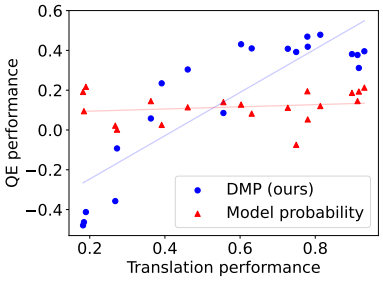


Figure 4: Relationship between model translation performance and QE performance.

Looking at Figure 4, DMP's QE performance is better for higher performing models, while model probability QE performance is more consistent across different models (but the performance is poor). This is somewhat expected, since the motivation of DMP 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 some output by the worse-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 5. 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 performance.

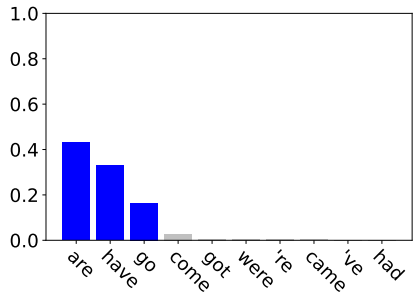


Figure 5: Example of Whisper Tiny's hallucinated probability distribution at an output step.

5.4 Finding Dominant Cluster

We experiment with common methods, originally used for sampling, to find the dominant tokens. Refer to Section 2.2 for an explanation of these methods. We use the same experiment setup as in Section 5.2.2: token-level quality estimation on the HJQE dataset. We use the HJQE development split to find the best hyperparameter for each setting, and apply them to the HJQE test split.

The results are shown in Table 6. Our method of finding dominant cluster performs generally better than the other methods, however, not by a large margin. Surprisingly, top-k performs quite well despite being a naive approach that always assumes the number of dominant tokens in a cluster to be fixed. The MCC score of Quality Estimation using top-k as dominant-cluster-finding method have almost the same performance as our approach. However, this might be due to the HJQE dev and test set being similar, thus tuning a good k value is enough to achieve good performance. Observe that, for top-k, the best k value for the *Scratch* model is $k = 2$, while for the *DeltaLM Large* model is $k = 5$. In contrast, the best hyperparameters found for other approaches are quite similar between the *Scratch* model and the *DeltaLM Large* model, indicating top-k is more sensitive to hyperparameters.

	HJQE dev	HJQE test	Best hyperparams *
Scratch			
top-k	0.199	0.130	k=2
ϵ-cut	0.197	0.128	ϵ=0.05
η-cut	0.169	0.108	η=0.1
top-p	0.134	0.077	p=0.9
min-p	0.169	0.109	p=0.9
difference-jump	**0.207**	**0.134**	x=30%, ϵ=0.01
DeltaLM			
top-k	**0.280**	0.147	k=5
ϵ-cut	0.254	0.149	ϵ=0.05
η-cut	0.219	0.124	η=0.1
top-p	0.153	0.088	p=0.7
min-p	0.234	0.137	p=0.9
difference-jump	**0.280**	**0.156**	x=30%, ϵ=0.1

* Best hyperparameters found on the dev split.

Table 6: Performance of DMP on token-level QE when using different methods for finding dominant clusters. We denote our method as "difference-jump".

6 Conclusion

In this paper, we first perform analysis showing the existence of dominant clusters with sizes larger than 1 in the model output probability distribution, which happens exclusively for x-to-many tasks. We show that the tokens in the dominant clusters are underconfident, as their probability is spread between other dominant options. Then, we proposed ***Dominant Mass Probability*** (DMP) - a Quality Estimation method that favors the dominant tokens to encounter generative models being underconfidence. Since DMP only utilizes the model probability distribution, it is low-cost, easy to implement, and can be applied to many model architectures. We show that DMP performs notably better than model probability, and better than probability entropy. For QE on Machine Translation, when using DMP on a translation model to evaluate other models' output, DMP is reaching close to the performance of supervised QE approaches.

Limitations

As discussed in Section 5.3, our method does not tackle cases where low-quality models are overconfident in their bad output. It's also unlikely to work for x-to-one text generation tasks like Automatic Speech Recognition, or multiple-choice Question-Answering, since the dominant clusters with sizes larger than 1 are unlikely to appear.

References

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.

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.

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.

Tu Anh Dinh, Tobias Palzer, and Jan Niehues. 2024. 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.

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, Chikiu 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. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

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.

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.

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, 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*.

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.

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.

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*.

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.

A 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.

B 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
- 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