Sigmoid Head for Quality Estimation under Language Ambiguity

Anonymous ACL submission

Abstract

Language model (LM) probability is not a reliable quality estimator, as natural language is ambiguous. When multiple output options are valid, the model's probability distribution is spread across them, which can misleadingly indicate low output quality. This issue is caused by two reasons: (1) LMs' final output activation is softmax, which does not allow multiple correct options to receive high probabilities simultaneously and (2) LMs' training data is single, one-hot encoded references, indicating that there is only one correct option at each output step. We propose training a module for Quality Estimation on top of pre-trained LMs to address these limitations. The module, called Sigmoid Head, is an extra unembedding head with sigmoid activation to tackle the first limitation. To tackle the second limitation, during the negative sampling process to train the Sigmoid Head, we use a heuristic to avoid selecting potentially alternative correct tokens. Our Sigmoid Head is computationally efficient during training and inference. The probability from Sigmoid Head is notably better quality signal compared to the original softmax head. As the Sigmoid Head does not rely on human-annotated quality data, it is more robust to out-of-domain settings compared to supervised QE.

1 Introduction

Quality Estimation (QE) is the task of providing a score estimation of the model output quality during inference when the ground-truth output is not available. The most straightforward way is to use model probability: it is computationally lightweight, does not require any additional modules, as the model probability comes for free during generation. However, previous works have shown that, for language generation tasks, model probability is not a good signal of output quality, largely due to the inherent ambiguity of natural language (Ott et al., 2018; Stahlberg and Kumar, 2022; Fadeeva et al., 2024; Flores et al., 2025; Dinh and Niehues, 2025). For a given input, multiple outputs can be valid, making the model probability spread out more even when the output is high quality. For convenience, we call this the ***ambiguity-induced underconfidence*** issue.

Ambiguity-induced underconfidence is partly caused by the architecture and training setup of current language models (LMs). First, the LMs' final output activation is softmax, which enforces the probabilities of all output options to sum to one. Second, the training target is a single reference that is one-hot encoded, which teaches the model that only one output option is correct. As a result, multiple valid options cannot all receive high probabilities at the same time.

We propose to address the above issue as follows: we train an additional unembedding layer (a.k.a. *"projection layer"* or *"LM head"*) on top of pre-trained LMs to perform Quality Estimation. The output activation of this layer is sigmoid instead of softmax, so each output option is modeled independently. We use the same training data used for the original LM to train our module. To address the disadvantage of the single-reference, one-hot encoding setup, we propose a heuristic for negative sampling during training to avoid selecting potentially alternative correct tokens as negative. We refer to the module as the **Sigmoid Head**.

In short, our contributions are as follows:

- We identify LMs' architecture and training issues causing ***ambiguity-induced underconfidence***.
- We propose a Sigmoid Head [1] on top of pre-trained LMs for Quality Estimation to address these issues. The module is trained on the same data as standard generative training, requires no additional labeled data, and is computationally lightweight during training and inference.
- Our experiments show that the probabilities from the Sigmoid Head provide better quality signals

[1] Code submitted as zip file.

than those from the standard softmax head. Moreover, since our method does not rely on human-labeled quality data for training, it is more robust and outperforms the supervised COMET Kiwi (Rei et al., 2022) on domain-specific machine translation (biomedical). It also outperforms common QE approaches: Monte Carlo Sequence Entropy (Malinin and Gales, 2021; Kuhn et al., 2023) and LLM Self Judge.

2 Background and Motivation

2.1 Background: Standard LM Training

Training Objective Text-generation language models (LMs) are trained auto-regressively. Given an input sequence $\mathbf{x} = (x_1, \ldots, x_{|\mathbf{x}|})$ and a previously generated output prefix $\mathbf{y}_{<i} = (y_1, \ldots, y_{i-1})$, the model with parameters θ is trained to predict the next token at time step i:

$$P_\theta(y_i \mid \mathbf{x}, \mathbf{y}_{<i}) \xrightarrow{\text{train}} \hat{P}(y_i \mid \mathbf{x}, \mathbf{y}_{<i}),$$

where the target distribution $\hat{P}$ is a one-hot vector that assigns probability 1 to the single reference token and 0 to all others. We exclude label smoothing from our explanation for simplicity (See Appendix A for more details).

Model Architecture Most text-generation models consist of a transformer (encoder-decoder or decoder-only) with parameters θ_{tr}, which produces a hidden representation $\mathbf{h}_i \in \mathbb{R}^d$ at generation step i. This (last) hidden state is mapped to vocabulary-sized logits $\mathbf{z}_i^{\text{out}}$ via an unembedding head with parameters θ_{out}:

$$\mathbf{z}_i^{\text{out}} = W_{\text{out}}\mathbf{h}_i + \mathbf{b}_{\text{out}},$$

where $W_{\text{out}} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $\mathbf{b}_{\text{out}} \in \mathbb{R}^{|\mathcal{V}|}$ ($\mathbf{b}_{\text{out}}$ is optional). A softmax function is then applied to obtain a distribution over the vocabulary $\mathcal{V}$:

$$P_\theta(y_i \mid \mathbf{x}, \mathbf{y}_{<i}) = \text{softmax}(\mathbf{z}_i^{\text{out}}),$$

where $\theta = \{\theta_{\text{tr}}, \theta_{\text{out}}\}$.

2.2 Motivation: Issues of Standard LMs

Natural language is ambiguous, where there are often multiple ways to express the same meaning. As a result, there are usually multiple correct output options at each generation step. The standard training above does not faciliate this in two aspects:

- **D1**: The softmax function enforces mutual exclusivity, i.e., probabilities must sum to one. When multiple tokens are correct, they cannot all receive high probability.
- **D2**: Training relies on a single one-hot reference at each step, teaching the model that exactly one token is correct and all others are incorrect.

While **D1** could be addressed by replacing the final softmax function with sigmoid, **D2** remains problematic. To further demonstrate **D2**, consider the following example where there are 2 samples in a translation training dataset (without loss of generality to other text-generation tasks such as instruction following or language modeling):

Input: Ich werde anfangen
Output 1: I will *begin*
Output 2: I will *start*

For the same input and output prefix, the model receives conflicting one-hot encoded target supervision. Sample 1 provides the target:

$$\hat{P}(y_i = \text{“begin”} \mid \mathbf{x}, \mathbf{y}_{<i}) = 1,$$
$$\hat{P}(y_i \neq \text{“begin”} \mid \mathbf{x}, \mathbf{y}_{<i}) = 0,$$

while Sample 2 provides:

$$\hat{P}(y_i = \text{“start”} \mid \mathbf{x}, \mathbf{y}_{<i}) = 1,$$
$$\hat{P}(y_i \neq \text{“start”} \mid \mathbf{x}, \mathbf{y}_{<i}) = 0.$$

Sample 1 encourages the model to place all probability mass on “begin” and none on “start”, while Sample 2 does the opposite. If the model assign probability close to 1 to “begin”, the loss will be low for Sample 1 but very high for Sample 2, where “begin” is an incorrect target. Similar argument holds for “start”. To reduce the total loss, the model cannot assign a probability close to 1 to either token. Instead, it must distribute probability mass between them. This applied to both models with softmax final activation and models with sigmoid final activation. See Appendix A for detailed behavior of the cross entropy loss in this example.

Implicit Effect of Ambiguity Language ambiguity is learned implicitly by language models. As discussed above, when the training data shows that multiple outputs are correct, the model learns to spread probability mass across these tokens. This behavior was also observed by Dinh and Niehues (2025). They show that, specifically for ambiguous generation tasks, multiple tokens often receive relatively high probability compared to the remaining tokens at each output step. They refer to these tokens as "dominant tokens", which are likely to correspond to valid alternative outputs. They propose a heuristic to identify such tokens. We adopt this idea in our method, described as follows.

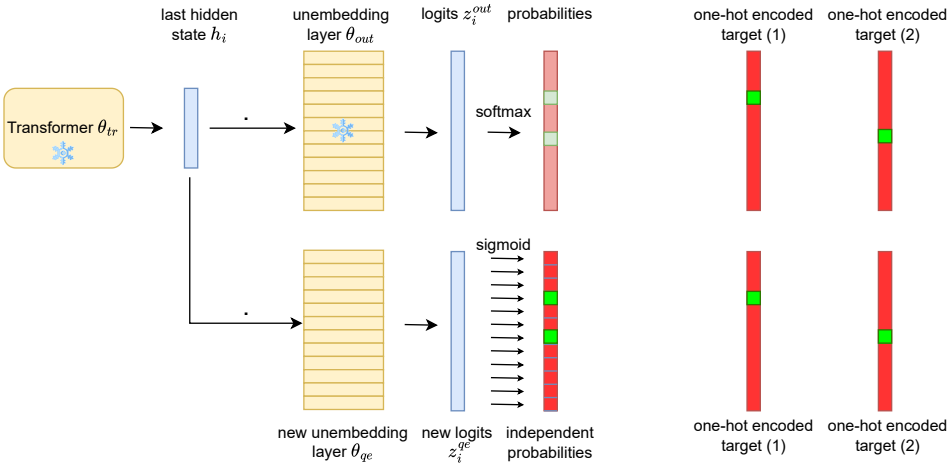


Figure 1: Extended language model (LM) architecture with our proposed Sigmoid Head. The weights of the original components from the LM are kept unchanged. We initialize the Sigmoid Head from the original softmax head, and train it to predict independent scores for each token in the vocabulary.

3 Proposed: Sigmoid Head for QE

We propose a lightweight module for Quality Estimation on top of a trained generative model that explicitly accounts for ambiguity in the training data. The weights $\theta = \{\theta_{\text{tr}}, \theta_{\text{out}}\}$ of the generative model are kept unchanged. We introduce an additional unembedding head with parameters θ_{qe}, operating on the same hidden states $\mathbf{h}_i$ produced by the transformer θ_{tr}, and trained on the same single-reference, one-hot encoded target as the standard setup of the original model (Figure 1).

Sigmoid Instead of Softmax To address the disadvantage **D1**, for our module, instead of using softmax as the output activation, we use the sigmoid function, similar to Noise-Contrastive Estimation (NCE) (Mnih and Teh, 2012). We then refer to our module as the **Sigmoid Head**. Concretely, given the transformer hidden state $\mathbf{h}_i$, the new unembedding head θ_{qe} computes the logits

$$\mathbf{z}_i^{\text{qe}} = W_{\text{qe}}\mathbf{h}_i + \mathbf{b}_{\text{qe}},$$

where $W_{\text{qe}} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $\mathbf{b}_{\text{qe}} \in \mathbb{R}^{|\mathcal{V}|}$ (optional) are trainable parameters. An element-wise sigmoid function is then applied:

$$P_{\theta'}(y_i \mid \mathbf{x}, \mathbf{y}_{<i}) = \sigma(\mathbf{z}_i^{\text{qe}})$$

where $\theta' = \{\theta_{\text{tr}}, \theta_{\text{qe}}\}$.

Unlike softmax, sigmoid removes the constraint that probabilities must sum to one. Multiple tokens can simultaneously receive high probability, better matching settings where multiple outputs are correct.

At each output step i, we train the Sigmoid Head to distinguish correct from incorrect tokens, given the one-hot encoded target. Let y_i^* denote the single reference token in the one-hot encoded target. The reference token is treated as a positive example with label 1. Negative examples are sampled from the non-reference tokens in the vocabulary $\mathcal{V} \setminus \{y_i^*\}$. We sample 10 negative tokens, as prior work has shown that this is an effective choice for different tasks (Mikolov et al., 2013, 2018). Negative tokens are sampled from a configurable distribution. Unless stated otherwise, we use token-frequency-based sampling, where tokens that appear more often in the training data are more likely to be selected as negatives. In this way, a token is expected to be selected as a negative as often as it appears as a positive. This balances the training signal and reduces bias toward frequent tokens. In Section 5.2, we evaluate alternative negative sampling strategies and analyze their impact on performance.

Ambiguity-Informed Negative Sampling Recall disadvantage **D2**, where some non-reference tokens from $\mathcal{V} \setminus \{y_i^*\}$ may be valid output alternatives. To reduce the risk of sampling such tokens as negatives, we make use of the observation from Dinh and Niehues (2025): in a trained LM, the tokens with dominant probability mass in the softmax distribution are likely to be the alternative correct options. We therefore apply their heuristic (See Appendix B for details) to identify the set of dominant tokens $\mathcal{D}_i$ from the original softmax distribution $P_\theta(y_i \mid \mathbf{x}, \mathbf{y}_{<i}) = \text{softmax}(\mathbf{z}_i^{\text{out}})$ and exclude

them from negative sampling. Our approach can be viewed as a form of implicit knowledge distillation, as we are using the knowledge from the pretrained transformer θ_{tr} and softmax head θ_{out} to train our new sigmoid head θ_{qe}. Formally, during training of the sigmoid head, negative samples $\mathcal{N}_i$ are drawn from

$$\mathcal{V} \setminus (\{y_i^*\} \cup \mathcal{D}_i)\,,$$

Intuitively, for potentially correct tokens in $\mathcal{D}_i$, the learning signal is postponed: the model neither learn that they are correct nor incorrect at this step.

The Sigmoid Head is trained with binary cross-entropy loss over the selected samples, consisting of the reference token y_i^* and the negative samples $\mathcal{N}_i$. The loss at output step i is:

$$\mathcal{L}_i = -\log p_i(y_i^*) - \sum_{v \in \mathcal{N}_i} \log\big(1 - p_i(v)\big),$$

where $p_i(v)$ is the probability that the Sigmoid Head assigned to token $v \in \mathcal{V}$ at time step i.

During inference, the output text is generated by the original LM components $\theta = \{\theta_{\text{tr}}, \theta_{\text{out}}\}$ without modification. The Sigmoid Head θ_{qe} operates in parallel by taking the last hidden states from θ_{tr} at each generation step and produces a quality score for each output token.

Our Sigmoid Head has several advantages:

- We use the same training data as standard generative modeling and require no extra annotations. Since we do not rely on human-annotated quality scores, unlike supervised QE, our approach is expected to be more robust across domains.
- Training is computationally efficient thanks to negative sampling: at each generation step, the loss is computed over the reference token and a small set of sampled negative tokens, rather than over the full vocabulary. Consequently, only the corresponding rows of the unembedding matrix are updated. This efficiency is similar to NCE, although in our case it is not the main motivation.
- Inference is computationally efficient, as the computation of the final hidden state is shared between the softmax head and the Sigmoid Head.

4 Experimental Setup

4.1 Models and Training Data

Details of the language models (LMs) used in our paper and their corresponding training data can be found in Table 1. We train Transformer Base from scratch using the base configuration of Fairseq (Ott et al., 2019) on 5M samples of ParaCrawl English-German Machine Translation data, filtered by Bicleaner AI (Zaragoza-Bernabeu et al., 2022; de Gibert et al., 2024). DeltaLM is finetuned on the same ParaCrawl data. Tower Instruct v2 and OLMo SFT v2 models are taken off the shelves. We experiment with adding our Sigmoid Head for Quality Estimation onto these models. More details on our motivation for model choices are in Appendix C.

Model	Nr. Params	Train Data*
Transformer Base (Ott et al., 2019)	0.062B	ParaCrawl en-de 5M (Bañón et al., 2020)
DeltaLM (Ma et al., 2021)	0.830B	ParaCrawl en-de 5M (Bañón et al., 2020)
Tower Instruct v2 (Alves et al., 2024)	7B	TowerBlocks v2 (Alves et al., 2024)
OLMo SFT v2 (OLMo et al., 2024)	1B	Tulu 3 SFT (Lambert et al., 2024)

*: Training data from the last training phase of the model.

Table 1: Models used in our paper, along with their training data from the last training phase. We use the same data to train our Sigmoid Heads.

4.2 Test Data and Evaluation

We evaluate the Quality Estimation performance of our Sigmoid Head on three different generation tasks: Machine Translation, Paraphrase, and Question Answering. We place more focus on the Machine Translation (MT) task, as MT evaluation is more well-studied, with different established data, models, and benchmarks thanks to the WMT Shared Tasks (Kocmi et al., 2024; Freitag et al., 2024; Zerva et al., 2024).

Machine Translation For MT, we use ParaCrawl and WMT22 test sets for the two self-trained models as they are on the sentence level. We use WMT24 and the domain-specific BioMQM test sets (Zouhar et al., 2024) for the other models. We evaluate two QE settings: using the LM to produce a quality score for its own translation, which we denote as "Eval Self", and using the LM to force-decode other translations and produce quality scores for the other translation, which we denote as "Eval Other". Human quality annotations are used as the ground-truth when available (Eval Other). Otherwise, for Eval Self, we use pseudo ground-truth produced by the reference-based XCOMET (Guerreiro et al., 2024), as Dinh et al. (2024) has shown that reference-based methods are robust enough to rank reference-free QE. We report Pearson correlation, which measures how the QE scores

linearly correlate with the ground truth quality.

Other Tasks For paraphrasing, we use the PAWS-X dataset (Yang et al., 2019) and evaluate both Eval Self and Eval Other settings. For question answering, we use GSM8k (Math domain) (Cobbe et al., 2021) and TruthfulQA (generic domain) (Lin et al., 2022) and only evaluate Eval Self. For Eval Self, pseudo ground-truth is generated by prompting Qwen2.5 72B Instruct (Team, 2024) to evaluate whether the model output is correct or not. In these tasks, ground-truth or pseudo ground-truth labels are binary, thus we report on Binary Cross Entropy (BCE) loss, which measures how close the QE scores are to the ground-truth binary labels, without the need to define a threshold.

More details on the test sets and choice of language pairs can be found in Appendix D.

Baselines We compare our approach to the standard softmax probability and BoostedProb (Dinh and Niehues, 2025). BoostedProb is an adjusted version of the softmax probability, which partially addresses *ambiguity-induced underconfidence*. Specifically, the tokens with dominant probability mass in the softmax distribution are assumed to be the likely correct output alternatives. The confidence of these tokens is then boosted by having the total probability mass of the whole dominant set as the quality score.

For MT, we use the supervised COMET Kiwi (Rei et al., 2022) as an upper baseline, as it is trained on human-labeled quality data as opposed to our approach. We also compare our approach to common unsupervised QE approaches. The first one is Monte Carlo sequence entropy (Malinin and Gales, 2021; Kuhn et al., 2023), where we sample 10 output sequences for each input and compute sequence-level probability entropy. The second one is LLM Self Judge, in which we prompt the model to evaluate its own output.

5 Results and Discussion

5.1 Overall Performance

QE performance on self-generated translation output (Eval Self) Table 2 shows the QE results on self-generated outputs from the Paracrawl, WMT22, and WMT24 translation datasets across different models and language pairs. Overall, our Sigmoid Head consistently outperforms the standard softmax head and the BoostedProb baseline. In many cases, it still underperforms the supervised COMET Kiwi. However, COMET Kiwi results might be biased, as we use XCOMET outputs as the gold quality scores, which belong to the same model family as COMET Kiwi.

	Softmax Head	Boosted Prob	Sigmoid Head	COMET Kiwi
TransformerBase				
ParaCrawl	0.155	0.235	**0.383**	**0.536**
WMT22 en-de	0.199	0.367	**0.586**	**0.722**
DeltaLM				
ParaCrawl	0.131	0.218	**0.403**	**0.537**
WMT22 en-de	0.165	0.291	**0.515**	**0.634**
Tower				
WMT 24 en-de	0.148	0.414	**0.468**	**0.562**
WMT 24 en-fr	0.155	0.370	**0.391**	**0.530**
WMT 24 en-es	0.183	**0.446**	0.415	**0.525**
WMT 24 en-pt	0.150	0.458	**0.472**	**0.533**
WMT 24 en-nl	0.145	0.419	**0.513**	**0.552**
WMT 24 en-it	0.154	0.394	**0.448**	**0.588**
WMT 24 en-ko	0.165	0.595	**0.566**	**0.626**
WMT 24 en-cn	0.160	0.491	**0.537**	**0.500**
WMT 24 en-ru	0.176	0.505	**0.524**	**0.625**
Olmo				
WMT 24 en-de	0.195	0.398	**0.606**	**0.588**
WMT 24 en-es	0.209	0.508	**0.672**	**0.613**

Best scores are shown as **<score>**, second best are **<score>**.

Table 2: QE performance on models' self-generated translations (Eval Self) in Pearson Correlation ↑.

QE performance on others' translation output (Eval Others) The QE results on scoring others' translations are shown in Table 3. As before, the Sigmoid Head outperforms the standard softmax head and the BoostedProb baseline, but still lags behind the supervised COMET Kiwi model on the WMT test sets. However, COMET Kiwi is trained on in-domain WMT shared task data, which gives it an advantage over our unsupervised approach. This is confirmed by the results on BioMQM, a domain-specific test set, where our method generally outperforms COMET Kiwi, indicating better robustness on out-of-domain settings.

Our approach also works on providing QE scores on the word level, allowing DeltaLM to outperform supervised QE. Details are in Appendix E.

QE performance on paraphrasing and question answering Table 4 shows the QE performance on tasks other than translations: paraphrasing and question answering. Our Sigmoid Head consistently outperforms Softmax Head and BoostedProb across different languages and test sets.

Comparison to common QE approaches Table 5 compares the QE performance of our Sigmoid Head with Monte Carlo Sequence Entropy

and LLM Self Judge. Monte Carlo Sequence Entropy is usually computed from log probabilities produced by the standard softmax head. We also report a variant that uses log probabilities from our Sigmoid Head. Overall, our Sigmoid Head consistently outperforms both the standard Monte Carlo Sequence Entropy baseline and the LLM Self Judge baseline. The Monte Carlo Sequence Entropy variant based on the Sigmoid Head either slightly degrades performance or gives negligible improvement over using the Sigmoid Head alone, while being more expensive because it requires multiple sampled sequences. LLM Self Judge performs poorly in most cases. This is because Tower is a specialized model that is not trained for general LLM-as-a-Judge tasks beyond machine translation evaluation, and OLMo lacks strong LLM-as-a-Judge capability due to its small size. The only exception is Tower Self Judge on the WMT24 translation data, since the model is trained for this task. These results highlight a limitation of LLM Self Judge: it does not work well in general unless the model is sufficiently large and not overly specialized for a small set of tasks.

	Softmax Head	Boosted Prob	Sigmoid Head	COMET Kiwi
TransformerBase				
WMT 22 en-de	0.076	0.131	**0.166**	**0.365**
DeltaLM				
WMT 22 en-de	0.081	0.141	**0.201**	**0.365**
Tower				
WMT 24 en-de	0.094	0.215	**0.272**	**0.349**
WMT 24 en-es	0.056	0.170	**0.230**	**0.429**
WMT 24 en-zh	0.011	0.114	**0.161**	**0.385**
WMT 24 en-ru	0.072	0.180	**0.205**	**0.345**
Average	0.058	0.170	**0.217**	**0.377**
BioMQM en-pt	0.067	**0.214**	**0.263**	0.078
BioMQM en-es	0.150	**0.300**	**0.314**	0.150
BioMQM en-fr	0.145	**0.364**	**0.369**	0.238
BioMQM en-de	0.114	**0.450**	**0.441**	0.194
BioMQM en-it	0.093	**0.343**	**0.268**	0.235
BioMQM en-zh	0.055	0.223	**0.316**	**0.274**
BioMQM en-ru	0.065	**0.301**	**0.300**	0.207
Average	0.098	**0.314**	**0.325**	0.197

Best scores are shown as **<score>**, second best are **<score>**.

Table 3: QE performance when forced-decoding on other translations (Eval Others) in Pearson ↑.

5.2 Effect of Negative Sampling Strategies

We perform an ablation study to see how different negative sampling strategies affect the QE performance of the Sigmoid Head. We run these experiments on Transformer-base and DeltaLM models, as these models allow multiple training runs with reasonable computational cost. Figure 2 plots ground-truth quality scores against predicted quality scores for different model settings.

Mode	Model	Lang.	Softmax Head	Boosted Prob	Sigmoid Head
Paraphrasing (Pawsx)					
Eval Self	Tower	en	11.668	1.925	**1.056**
		de	9.025	1.645	**0.422**
		es	9.334	1.362	**0.399**
		fr	8.847	1.256	**0.325**
		zh	12.745	2.925	**0.622**
	Olmo	en	14.684	3.769	**0.883**
Eval Others	Tower	en	10.835	6.858	**4.546**
		de	11.756	8.726	**3.151**
		es	10.868	7.324	**2.309**
		fr	9.977	6.297	**1.849**
		zh	12.362	9.257	**2.172**
	Olmo	en	13.364	9.096	**2.832**
Question Answering Math (GSM8K)					
Eval Self	Olmo	en	12.802	0.843	**0.642**
Question Answering Generic (TruthfulQA)					
Eval Self	Olmo	en	9.424	6.219	**1.698**

Table 4: QE performance on tasks other than translation, meansured in Binary Cross Entropy loss (BCE) ↓.

Test Data	Model	Sigmoid Head	Monte Carlo Seq. Entropy		LLM Self Judge
			Softmax	Sigmoid	
QE performance in Pearson ↑					
WMT24	Tower	**0.468**	0.093	0.441	0.293
	Olmo	**0.606**	0.106	0.552	0.093
QE performance in BCE ↓					
PAWS-X	Tower	1.056	11.828	**1.047**	22.993
	Olmo	**0.883**	16.228	0.920	6.416
GSM8k	Olmo	**0.642**	15.241	0.798	16.408
TruthfulQA	Olmo	1.698	9.681	**1.674**	24.729

Table 5: Comparison to common QE approaches.

Standard Softmax: Underconfidence Figure 2a shows the behavior of the standard softmax head. We observe the *ambiguity-induced underconfidence* issue: outputs with high gold quality (points on the right side of the plot) have predicted probabilities spread across the full range from 0 to 1.

Random Negative Sampling Figure 2b shows the Sigmoid Head trained with random negative sampling. The probability is overconfident: outputs of all quality levels receive probabilities close to 1. This happens for two main reasons. First, at each step, incorrect tokens greatly outnumber correct ones. Random sampling therefore mostly picks clearly wrong tokens, which gives a weak

		Transformer Base		DeltaLM	
		ParaCrawl	WMT22	ParaCrawl	WMT22
Probability (Softmax Head)		0.155	0.199	0.131	0.165
BoostedProb		0.235	0.367	0.218	0.291
	Negative sampling				
Sigmoid Head	Random	0.123	0.163	0.141	0.209
	Token Freq	0.304	0.521	0.142	0.209
	Token Freq + Avoid Dominant	0.380	0.588	0.394	0.508
	Softmax	0.197	0.441	0.126	0.357
	Softmax + Avoid Dominant	0.326	0.580	0.239	0.467
	Softmax t2 + Avoid Dominant	0.373	**0.602**	0.387	0.510
	Softmax t2 + Token Freq + Avoid Dominant	**0.383**	0.586	**0.403**	**0.515**
COMET Kiwi		**0.536**	**0.722**	**0.537**	**0.634**

Best scores are shown as **<score>**, second best are **<score>**.

Table 6: Effect of different negative sampling strategies on QE performance.

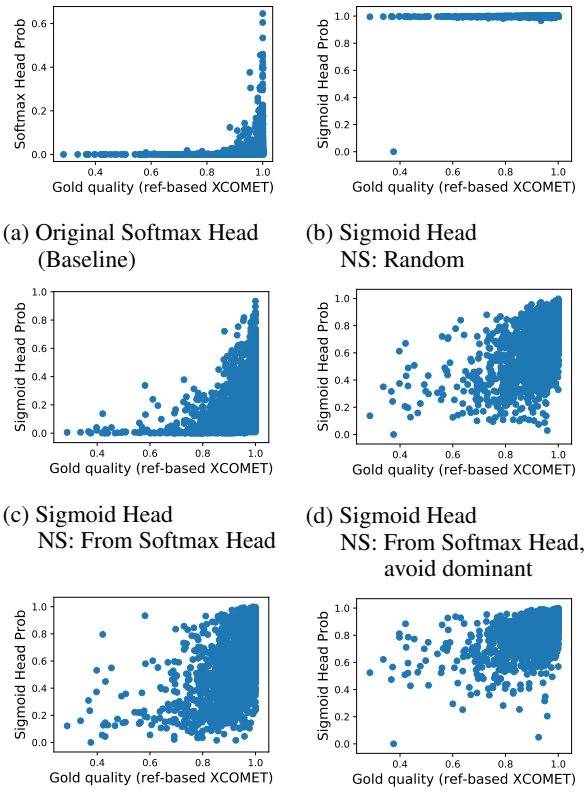
learning signal and does not teach the model to distinguish correct tokens from plausible but incorrect ones. Second, random sampling treats all tokens as negatives equally often, while positive examples depend on token frequency in the training data. As a result, frequent tokens appear more often as positives than as negatives, biasing the model to assign them high scores even when they are incorrect.

Negative Sampling from the Softmax Head Figure 2c shows the Sigmoid Head trained with negative sampling from the softmax head distribution, where highly ranked tokens (except the reference) are more likely to be sampled as negatives. In this case, the sigmoid head remains underconfident, similar to the baseline in Figure 2a, although the effect is weaker. This is because alternatively correct tokens are still likely to be sampled as negatives, which exposes the model to the same *ambiguity-induced underconfidence* issue.

Avoiding Dominant Tokens The *ambiguity-induced underconfidence* issue is mitigated when we explicitly exclude dominant tokens from negative sampling. As shown in Figure 2d, the predicted probabilities better align with gold quality.

Negative Sampling from Token Frequency Similar trends are observed when negative sampling is based on token frequency in the training data. Figure 2e shows that frequency-based sampling alone still suffers from underconfidence. When dominant tokens are explicitly excluded (Figure 2f), the issue is mitigated, and the confidence estimates become more aligned with gold quality.

Quantitative Results The above observations are confirmed by the QE performance results in terms of Pearson correlation, shown in Table 6.



(a) Original Softmax Head (Baseline)
(b) Sigmoid Head NS: Random
(c) Sigmoid Head NS: From Softmax Head
(d) Sigmoid Head NS: From Softmax Head, avoid dominant
(e) Sigmoid Head NS: Token Frequency
(f) Sigmoid Head NS: Token Frequency, avoid dominant

Figure 2: Ground-truth versus predicted quality scores.

The proposed Sigmoid Head consistently outperforms the standard softmax head and the BoostedProb baseline. The best-performing settings use negative sampling based on token frequency or the softmax distribution, combined with explicit exclusion of dominant tokens. We further experiment with applying a temperature of 2 to the softmax distribution used for negative sampling. This reduces the impact of *ambiguity-induced underconfidence*

and leads to the best overall results. This best setting is used for Transformer Base and DeltaLM in the main experiments (Section 5.1),

For Olmo and Tower, we only evaluate a small number of top settings identified from the Transformer-based and DeltaLM experiments, due to their higher computational cost for training. Our proxy runs show that negative sampling from token frequency alone, with dominant tokens excluded, performs best for these models. We therefore use this setting in the main experiment (Section 5.1) for OLMo and Tower.

6 Related Work

In the field of Machine Translation (MT), the Quality Estimation task is well studied, with the well-established method of training a separate module on human-labeled quality scores on MT output, with the representative example of COMET-kiwi (Rei et al., 2022). However, outside of MT, there is a lack of such training data to build such supervised QE modules. This calls for the need of unsupervised QE approaches, which are well aligned with *Uncertainty Quantification* techniques. These techniques often fall into several categories (Fadeeva et al., 2023): information-based approaches which make use of the model probability (Fomicheva et al., 2020), ensemble-based approaches which require generating multiple outputs (Kuhn et al., 2023; Dinh and Niehues, 2023), self-validation approaches which prompt the LLMs to evaluate themselves (Kadavath et al., 2022), and density-based approaches which require access to the model training data to detect out-of-distribution instances during inference (Lee et al., 2018; Ren et al., 2023). Amongst these categories, information-based approaches are the most straightforward and computationally lightweight, as model probabilities come for free at generation.

Previous works have pointed out that model probability for text generation tasks is not a good signal of output quality due to the inherent ambiguity of natural language, which we called the **ambiguity-induced underconfidence** issue (Ott et al., 2018; Stahlberg and Kumar, 2022; Fadeeva et al., 2024; Flores et al., 2025; Dinh and Niehues, 2025). Fadeeva et al. (2024) address this issue by detecting tokens that are synonyms and contradicting compared to the final selected token, and re-calculate the quality score given the probabilities of only those tokens. Flores et al. (2025) address the issue by, instead of only looking at the probability of the output as a quality signal, they calculate the probability ratio between the highest-ranked sequences and the rest, or evaluate the thinness of the distribution's tail. Dinh and Niehues (2025) addresses the issue by assigning dominant tokens in the softmax distribution a quality score equal to the total probability mass of all dominant tokens, rather than their individual probability mass. These works address *ambiguity-induced underconfidence* by modifying the softmax probability distributions.

In this paper, we address *ambiguity-induced underconfidence* at the training architectural level: instead of softmax, we use the sigmoid activation, so that each output option is modeled independently. In this regard, noise contrastive estimation (NCE) (Gutmann and Hyvärinen, 2010; Mnih and Teh, 2012) and SCONES (Stahlberg and Kumar, 2022) are the closest to our work, since they also switch out the softmax activation for sigmoid. However, NCE's focus is to train models on a large vocabulary size more efficiently with negative sampling so that not the whole unembedding layer needs to be learnt at every update step; SCONES' focus is to improve generation quality. Unlike us, both works do not take into account the *ambiguity-induced underconfidence* issue, i.e., all non-reference options can still be sampled as negative, even when they could be the alternative correct options.

7 Conclusion

This work addressed the *ambiguity-induced underconfidence* issue of LM probability for QE by introducing the Sigmoid Head, an additional unembedding head with sigmoid activation trained on top of frozen pretrained models. The sigmoid activation allows multiple output options to receive high confidence, while ambiguity-informed negative sampling avoids penalizing likely alternative correct tokens during training. The Sigmoid Head is trained on the same data as standard language modeling, requires no human-annotated quality labels, and is computationally efficient at both training and inference time. Across machine translation, paraphrasing, and question answering, the probability from Sigmoid Head provides a stronger quality signal than the standard softmax head and common unsupervised QE baselines. As it does not rely on supervised quality data, it is also more robust in out-of-domain settings and can outperform supervised QE models under domain shift.

Limitations

The main motivation of this work is to improve the training setup and model architecture to better account for the ambiguity of natural language. In standard large language model training, such ambiguity already exists during next-token prediction pretraining, where multiple tokens can be valid at each generation step. However, due to limited computational resources, we only apply our proposed training setup to the final instruction-following training stage. Applying our approach during the pretraining phase could potentially bring stronger benefits, as the model could learn ambiguity more effectively from the large-scale pretraining data. Another limitation is that we evaluate our approach on only two LLMs, Tower Instruct 7B and OLMo 1B, as these are among the few model families for which the training data is publicly available.

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.

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, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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 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 and Jan Niehues. 2025. Are generative models underconfident? better quality estimation with boosted model probability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3364–3382, Suzhou, China. Association for Computational Linguistics.

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, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385, Bangkok, Thailand. Association for Computational Linguistics.

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.

Lorenzo Jaime Yu Flores, Ori Ernst, and Jackie CK Cheung. 2025. Improving the calibration of confidence scores in text generation using the output distribution's characteristics. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 172–182, Vienna, Austria. 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, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio,

Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? results of the WMT24 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81, Miami, Florida, USA. 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.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, and 3 others. 2024. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2302.09664*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training.

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.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

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*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *International Conference on Machine Learning*.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2024. 2 olmo 2 furious.

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

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*.

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.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André FT Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725.

Felix Stahlberg and Shankar Kumar. 2022. Jam or cream first? modeling ambiguity in neural machine translation with SCONES. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4950–4961, Seattle, United States. Association for Computational Linguistics.

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

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. 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.

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.

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 Cross-entropy Loss Behaviour on Multi-target Samples

Recall the example in Section 2.2 where there are 2 samples in the training data:

Input: Ich fange an
Output 1: I will *begin*
Output 2: I will *start*

which lead to the two training targets:

$$\hat{P}(y_i = \text{“begin”} \mid \mathbf{x}, \mathbf{y}_{<i}) = 1,$$
$$\hat{P}(y_i \neq \text{“begin”} \mid \mathbf{x}, \mathbf{y}_{<i}) = 0,$$

and

$$\hat{P}(y_i = \text{“start”} \mid \mathbf{x}, \mathbf{y}_{<i}) = 1,$$
$$\hat{P}(y_i \neq \text{“start”} \mid \mathbf{x}, \mathbf{y}_{<i}) = 0.$$

When the output activation function is the standard softmax, the loss commonly used in practice is referred to as the negative log-likelihood (NLL), which is equivalent to the categorical cross-entropy loss with one-hot target:

$$\mathcal{L}_{\text{CE}} = -\sum_{v}^{\mathcal{V}} t_v \log p_v,$$

where t_v is the one-hot target and p_v is the predicted probability. When $p_{\text{begin}} \approx 1$, p_{start} is enforced to be close to 0 given the softmax. The loss value is then low for Sample 1. However, for Sample 2 where $t_{\text{start}} = 1$, the loss value will be very large:

$$\begin{aligned}
\mathcal{L}_{\text{CE}} &= -\sum_{v}^{\mathcal{V}} t_v \log p_v \\
&= -\sum_{v \neq \text{start}}^{\mathcal{V}} t_v \log p_v - t_{\text{start}} \log p_{\text{start}} \\
&\approx -\sum_{v \neq \text{start}}^{\mathcal{V}} t_v \log p_v - 1 \times \log 0 \\
&\approx \infty.
\end{aligned}$$

Similarly for the case when the model assigns $p_{\text{start}} \approx 1$. In practice, label smoothing is often applied to alleviate this issue to some extent by redistributing a small amount of probability mass from the ground-truth token to other tokens in the target distribution. However, label smoothing does not fundamentally resolve the problem: it still enforces a single preferred target per sample and penalizes assigning high probability to alternative valid outputs, only less severely. As a result, the model is still discouraged from assigning high probability to multiple correct tokens simultaneously.

This issue persists even when we switch out the softmax output function with sigmoid. Consider the binary cross-entropy loss commonly used with the sigmoid output function:

$$\mathcal{L}_{\text{BCE}} = -\sum_{v}^{\mathcal{V}} \left[t_v \log p_v + (1 - t_v) \log(1 - p_v) \right].$$

Again, if $p_{\text{begin}} \approx 1$, the loss is low for Sample 1 but very high for Sample 2 where $t_{\text{begin}} = 0$:

$$\begin{aligned}
\mathcal{L}_{\text{BCE}} &= -\sum_{v}^{\mathcal{V}} \left[t_v \log p_v + (1 - t_v) \log(1 - p_v) \right] \\
&= -\sum_{v \neq \text{begin}}^{\mathcal{V}} \left[t_v \log p_v + (1 - t_v) \log(1 - p_v) \right] \\
&\quad - t_{\text{begin}} \log p_{\text{begin}} - (1 - t_{\text{begin}}) \log(1 - p_{\text{begin}}) \\
&= -\sum_{v \neq \text{begin}}^{\mathcal{V}} \left[t_v \log p_v + (1 - t_v) \log(1 - p_v) \right] \\
&\quad - 1 \times \log 0 \\
&\approx \infty.
\end{aligned}$$

Similarly for the case when the model assigns $p_{\text{start}} \approx 1$. Therefore, even with sigmoid, one-hot target supervision teaches the model to distribute probability mass amongst valid alternatives, preventing it from assigning high confidence to multiple correct tokens.

B Finding Dominant Tokens

Dinh and Niehues (2025) show that, at an output step, tokens with dominant probability mass in the softmax distribution (so-called “dominant tokens”) tend to be correct alternative outputs. They propose a heuristic to identify these tokens. First, the probability distribution is sorted in descending order. Then, moving from high to low probabilities, they search for a “significant drop.” A drop is considered significant if it exceeds both a relative threshold ($x = 30\%$ of the preceding probability) and an absolute threshold $\epsilon = 0.005$. This drop separates the dominant tokens from the remaining ones. In our work, we apply this heuristic to identify dominant tokens and exclude them from negative sampling, since they are potentially correct alternatives.

C Motivation for Model Choices

We selected four models to add the Sigmoid Head to: Transformer Base (trained from scratch), DeltaLM (fine-tuned), Tower, and OLMo (off-the-shelf). As mentioned in Section 4.2, we focus on Machine Translation (MT), since MT evaluation is better studied than evaluation for other tasks. We therefore trained the encoder-decoder Transformer Base and DeltaLM on a medium-sized MT dataset (ParaCrawl 5M) to enable faster experiments with different negative sampling methods. We chose Tower because it is a state-of-the-art decoder-only LLM specialized for MT. We chose OLMo, a general-purpose language model, to test whether the Sigmoid Head also works for generation tasks beyond MT.

An important factor is that we want to train our Sigmoid Head on the exact same data that was used to train the original language model. This is another reason for choosing Tower and OLMo, as their training data is publicly available. We use the instruction-tuned versions of these two models to reduce the computational cost of training the Sigmoid Head, since instruction-tuning data is typically much smaller than language modeling data.

D Test Sets for Evaluation

Table 7 shows the statistics of the test sets used in our experiments. More details of the test sets are as follows.

WMT and BioMQM The WMT and BioMQM machine translation test sets contain source sentences and reference translations for multiple language pairs. They also include system output translations from WMT shared task participants, together with human quality scores. We use these scores as ground truth for quality estimation, which enables evaluation in the *Eval Other* mode.

PAWS-X Each sample in PAWS-X is a pair of sentences (S_1, S_2) along with binary labels: 1 if the two sentences are paraphrases, 0 otherwise. This also enables our assessment of the "Eval Other" mode, where we treat S_1 as the input text, S_2 as the candidate output text, and the binary labels as the ground truth quality score. In contrast, for the "Eval Self" mode, we let the LM generate its own paraphrase of S_1, and generate pseudo ground truth quality score with Qwen2.5 72B Instruct.

GSM8k and TruthfulQA Each GSM8k example contains a math problem and its ground-truth answer. Each TruthfulQA example contains a question, a set of correct answers, and a set of incorrect answers. For these two datasets, we only evaluate the *Eval Self* mode. Pseudo ground-truth quality scores are generated using Qwen2.5 72B Instruct. To improve the reliability of the pseudo ground-truth, we provide all available reference information to Qwen2.5 72B Instruct: the ground-truth answer for GSM8k, and the correct and incorrect answer lists for TruthfulQA.

Language selection For the multilingual translation test sets (WMT24 and BioMQM), for Tower, we include all the available English-X language pairs that Tower was trained on: German, French, Spanish, Dutch, Italian, Korean, Chinese and Russian. For OLMo, since it is claimed to be English-centric and not meant for translation, we only include two high-resource language pairs: English-German and English-Spanish. Similarly, for the multilingual PAWS-X paraphrase test set, we evaluate all languages supported by Tower. For OLMo, we evaluate only English.

Task	Test Set	Lang.	Nr. In [1]	Nr. Out [2]
MT	ParaCrawl	en-de	5000	-
	WMT22	en-de	2037	10980
	WMT24	en-de	960	8262
		en-fr	960	-
		en-es	960	8242
		en-pt	960	-
		en-nl	960	-
		en-it	960	-
		en-ko	960	-
		en-cn	960	7608
		en-ru	960	8242
	BioMQM	en-pt	935	449
		en-es	717	2188
		en-fr	614	2456
		en-de	757	2710
		en-it	1040	514
		en-zh	639	4437
		en-ru	550	1650
Paraphrasing	PAWS-X	en	1749	1749
		de	1972	1972
		es	1976	1976
		fr	1958	1958
		zh	1962	1962
QA	GSM8k	en	1319	-
	TruthfulQA	en	817	-

[1]: Number of input samples, which is the test size for Eval Self.
[2]: Number of candidate outputs, which is the test size for Eval Others.

Table 7: Statistics of test sets.

E Word-level QE

We evaluate our Sigmoid Head on providing quality scores on the word level. We use the HJQE test set (Yang et al., 2023), which contains source sentences and participants' translations from the WMT20 QE Shared Task (Specia et al., 2020) along with human-annotated quality labels (OK/BAD) on the word level. We again do force-decoding to score these translations (Eval Others). As the labels are binary, we report Binary Cross Entropy loss (BCE). As an upper baseline, we use the supervised WMT 21 OpenKiwi model (Specia et al., 2021; Kim et al., 2017).

The results are shown in Table 8. Our Sigmoid Head again outperforms the Softmax Head and BoostedProb baselines. Sigmoid Head on top of the small Transformer Base model underperforms compared to the supervised QE. However, Sigmoid Head on top of DeltaLM enables DeltaLM to slightly outperform supervised QE, which was not the case for Softmax Head and BoostedProb.

Model	Softmax Head	Boosted Prob	Sigmoid Head	Supervised QE
Transformer Base	1.580	0.984	0.513	**0.332**
DeltaLM	1.362	0.829	**0.329**	0.332

Table 8: QE performance on word-level, measured in BCE loss ↓ (Eval Others).

F Hardware

Training the Sigmoid Head on top of LLMs (Tower and OLMo) is conducted on a single H100 GPU with 96 GB memory. All other processes are run on a single A100 GPU with 40 GB memory, including training the Transformer Base and DeltaLM models and performing inference for all models.

G License For Artifacts

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

- ParaCrawl dataset (Bañón et al., 2020): Creative Commons CC0
- WMT22 dataset (Kocmi et al., 2022): Apache License 2.0
- WMT24 dataset (Kocmi et al., 2024): Apache License 2.0
- BioMQM (Zouhar et al., 2024): Apache License 2.0
- GSM8k dataset (Cobbe et al., 2021): MIT License
- TruthfulQA dataset (Lin et al., 2022): Apache License 2.0
- DeltaLM model (Ma et al., 2021): MIT License
- Tower model (Alves et al., 2024): CC BY NC 4.0
- OLMo model (OLMo et al., 2024): Apache License 2.0