Don’t Be Overconfident When You’re Wrong; Don’t Be Underconfident When You’re Right

Anonymous ACL submission

Abstract

When responding to any question from the user or an API, a conversational search or question answering system should ideally be able to attach an appropriate confidence score to its output. While such systems are often overconfident, there are also situations where the system responds correctly yet lacks enough confidence. Underconfident responses cannot be relied upon, and therefore may not be utilised by the user or downstream tasks. Ideally, we want to know when systems are underconfident as well as when they are overconfident, and want to suppress both phenomena in a balanced manner. Furthermore, in this scenario, we want an evaluation measure that is guaranteed to (a) penalise a lowered confidence for a correct response; and also (b) penalise a raised confidence for an incorrect response. In light of this, we propose HMR (Harmonic Mean of Rewards) and demonstrate its advantages over existing calibration measures for our purpose by means of examples, axioms, and theorems.

1 Introduction

Large language models (LLMs) *hallucinate* (Ji et al., 2023), often with confidence. The system’s confidence about its own response may be given as an accompanying confidence score, or may be expressed in natural language (e.g., “`Yes I am certain.`” (Liu et al., 2023, Figure 9)). The former is particularly useful if the system response is going to be utilised for some downstream tasks: we can decide how much the upstream pieces of information can be relied upon based on the scores. Even if the system does not return a separate score that represents its internal confidence, a postprocessing step may be applied, where the input contains the system’s natural language response and the output is an *estimated* confidence score; the estimator may well be another LLM.

While *overconfidence* (i.e., the system returns an inaccurate response with high confidence) is a major problem, the other side of the coin is *underconfidence* (i.e., the system returns an accurate response but lacks confidence). If the system is underconfident, the user or the downstream tasks may not be able to utilise the correct response. To illustrate how both overconfident and underconfident cases may occur even for the same system, Figure 1 visualises the confidence scores we obtained in a pilot experiment using the SWAG dataset (Zellers et al., 2018), in which Google Bard served as an external confidence estimator for each pair of an *incomplete video caption* and either a *gold* final phrase (i.e., one that appropriately completes the caption) or a *distractor* final phrase (i.e., an unsuitable completion). Here, Google Bard was prompted to pretend that the final phrase provided to it was its own response in a caption completion task, and to return a confidence score for the response in the 0-1 range. Details are in Appendix A. Ideally, the confidence score for each gold response should be high, while that for each distractor response should be low.

In Figure 1, the red horizontal line indicates 50% confidence: for convenience, let us say for now that the system is *overconfident* if its confidence score for a distractor response is higher than 50%, (or exactly 50% which would also be practically inconvenient for the user or the downstream tasks); similarly, let us say that the system is *underconfident* if its confidence score for a gold response is lower than or exactly 50%. While there are 59 overconfident cases (out of 100 distractor responses), there are also 8 underconfident cases (out of 100 gold responses) as indicated by the red arrows. We argue that we should be able to make a distinction between overconfidence and underconfidence when evaluating a system like this, because remedying the two phenomena may require different approaches, and we do not necessarily want one of them suppressed at the expense of the other. Rather, we may want the system to balance the two. In this scenario, we want an evaluation measure that is

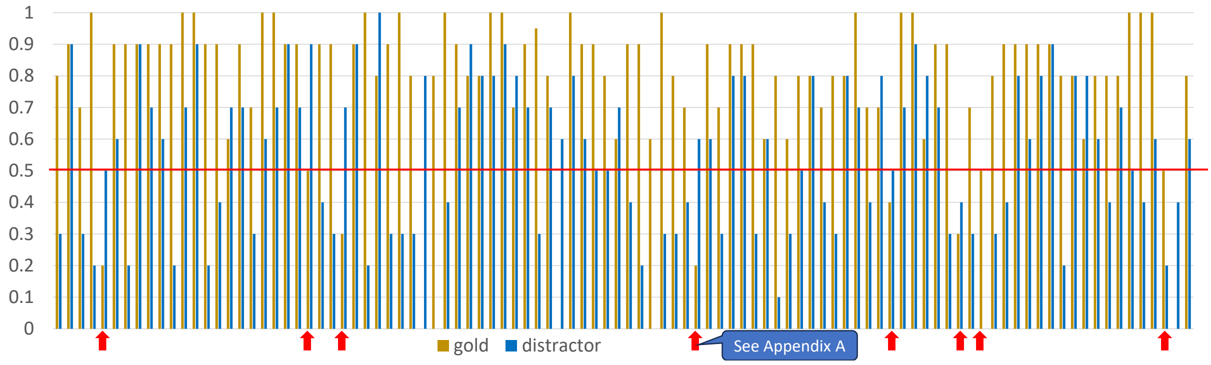


Figure 1: Results of a pilot experiment, in which a sample from the SWAG (Zellers et al., 2018) validation dataset was used to obtain confidence scores from Google Bard. The x axis represents the validation instances (with 100 gold and 100 distractor cases), and the y axis represents the confidence scores returned by Bard. The red arrows indicate underconfident cases, i.e., gold responses with low confidence scores. Details can be found in Appendix A.

guaranteed to (a) penalise a lowered confidence for a correct response; *and* (b) penalise a raised confidence for an incorrect response.

Calibration (Pakdaman Naeini et al., 2015) is the task of aligning confidence scores to the actual response accuracy. However, traditional measures used in calibration tasks only quantify how much the confidence scores deviate from the accuracy; they do not distinguish between overconfidence and underconfidence. We therefore propose a very simple and intuitive evaluation measure called *HMR* (Harmonic Mean of Rewards) and demonstrate its advantages over existing calibration measures for the purpose discussed above by means of examples, axioms, and theorems. More specifically, we show that while HMR possesses Properties (a) and (b) mentioned above, none of the calibration measures considered in this study do.

2 Prior Art

In calibration tasks, the *Expected Calibration Error* (ECE) is probably the most widely used evaluation measure. ECE is defined in Pakdaman Naeini et al. (2015), along with the *Maximum Calibration Error* (MCE). The premise is that we are given a set of instances, where each instance is associated with a binary gold label (i.e., correct or not) as well as a confidence score. In the context of a classification task with $M(\geq 2)$ classes (i.e., selecting a correct class or answer from M choices) , the confidence score may be the *top probability* (i.e., highest probability representing the most likely class/answer) of the set of M estimated correctness probabilities. To compute ECE or MCE, the N instances are first sorted by confidence scores, and are then partitioned into B *bins* for a given B, with the b-th bin containing n_b instances ($b = 1, \ldots, B$). For a given system that returned N responses along with confidence scores, let a_b denote the accuracy (i.e., fraction of correct responses) for Bin b,; let $\bar{c}_b$ denote the *average* confidence score for Bin b. Then ECE and MCE are given by:

$$\mathit{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} |\bar{c}_b - a_b| \ , \ \mathit{MCE} = \max_b |\bar{c}_b - a_b| \ . \tag{1}$$

Note that instance binning is a necessity for the introduction of the notion of binwise accuracy.

Two simple binning methods are commonly used in the literature: *equal width binning* (where the $[0, 1]$ range is partitioned into B bins of equal width) (Guo et al., 2017; Wang et al., 2023; Jiang et al., 2021; Portillo Wightman et al., 2023; Tam et al., 2023; Zablotskaia et al., 2023; Zhu et al., 2023) and *uniform mass binning* (n_b is the same for all bins) (Nguyen and O'Connor, 2015; Nixon et al., 2020; Lin et al., 2022; Liang et al., 2023). Kumar et al. (2020) discuss a theoretical advantage of uniform mass binning over equal width binning. Hereafter, we shall focus on uniform mass binning for convenience, but our findings on ECE and MCE do not depend on this choice.

One of the weaknesses of ECE and MCE is that they rely on the parameter B. Hence, we also discusses existing *binning-free* calibration measures.

Consider a classification task with $M(\geq\ 2)$ classes with N instances to classify; let $\mathit{GOLD}_j^m =$ 1 if Class m is the true class for the j-th instance, and 0 otherwise. For a classifier that returns M probabilities $(p_j^1, \ldots, p_j^M)$ s.t. $\sum_{m=1}^{M} p_j^m = 1$ for each instance, the Brier Score (Brier, 1950; Wang et al., 2023; Ovadia et al., 2019; Tian et al., 2023) may be applied:

$$BS = \frac{1}{N} \sum_{j=1}^{N} \sum_{m=1}^{M} (p_j^m - GOLD_j^m)^2 \ . \quad (2)$$

Brier proposed this measure in 1950 for verifying weather forecasts. To ensure a $[0, 1]$ range, we shall consider *Normalised* BS (NBS), which divides the sum in Eq. 2 by NM instead of N. However, BS is known to reflect classification errors as well as calibration errors (Gupta et al., 2021).

71 years later, Gupta et al. (2021) proposed a binning-free measure called *KS*, inspired by the Kolmogorov-Smirnov test for equality of two distributions (Hays, 1994). Given N confidence scores (e.g., top probabilities), let $(p_1, \ldots, p_N)$ be these scores after an ascending sort, and let $GOLD_j = 1$ if the instance that corresponds to the j-th score in the sorted list is correct, and 0 otherwise. Then,

$$cp_j = \frac{1}{N} \sum_{k=1}^{j} p_k \ , \quad cGOLD_j = \frac{1}{N} \sum_{k=1}^{j} GOLD_k \ , \quad (3)$$

$$KS = \max_j |cp_j - cGOLD_j| \ . \quad (4)$$

Recall that, in classification tasks with M classes, a system response may be associated with M probabilities rather than one confidence score; in principle, measures like ECE/MCE and KS may be applied to non-top probabilities as well. Some studies have in fact incorporated non-top probabilities in calibration evaluation (Vaicenavicius et al., 2019; Gupta et al., 2021; Nixon et al., 2020). However, our interest lies elsewhere: we want to evaluate overconfidence and underconfidence when each instance is associated with a binary gold label and *one* confidence score.

3 Proposed Evaluation Measures

We propose a very simple and interpretable binning-free evaluation approach that first quantifies overconfidence and underconfidence separately. For a given system, let I^- and I^+ denote the sets of instances for which the system's choices are considered incorrect and correct, respectively ($|I^-| + |I^+| = N$). Let $p(i)$ denote the system's confidence for Instance i. Then, for each $i \in I^-$ (the system is incorrect), $p(i)$ should be as close to 0 as possible; whereas for $i \in I^+$ (the system is correct), $p(i)$ should be as close to 1 as possible. Hence, we first define the *Rewards* for *suppressing* overconfidence and underconfidence separately as follows.

$$O = \sum_{i \in I^-} p(i) \ , \quad U = \sum_{i \in I^+} (1 - p(i)) \ , \quad (5)$$

$$R_O = \begin{cases} 1 & \text{if } I^- = \emptyset \ , \\ 1 - O/|I^-| & \text{otherwise} \ . \end{cases} \quad (6)$$

$$R_U = \begin{cases} 1 & \text{if } I^+ = \emptyset \ , \\ 1 - U/|I^+| & \text{otherwise} \ . \end{cases} \quad (7)$$

Note, for example, that when $I^- = \emptyset$ (i.e., all N system responses are correct), there is no way for the system to be overconfident for any of the instances and therefore $R_O = 1$ (i.e., perfection).

As we want systems to balance the above two rather than to sacrifice one for the sake of the other, let us consider the Harmonic Mean (Sakai, 2021):

$$HMR = \begin{cases} 0 & \text{if } R_O = R_U = 0 \ , \\ \frac{2\, R_O\, R_U}{R_O + R_U} & \text{otherwise} \ . \end{cases} \quad (8)$$

Note its advantage over the arithmetic mean. For example, consider two situations, $R_O = R_U = 0.5$ and $R_O = 0.9, R_U = 0.1$: the arithmetic means of R_O and R_U are the same, but $HMR = 0.500$ for the former and $HMR = 0.180$ for the latter.

Despite its simplicity, our measures are clearly advantageous over existing calibration measures for the purpose of penalising overconfidence and underconfidence separately, as we shall demonstrate below.

4 How the Measures Work (or Not)

In this section, we demonstrate how the proposed and existing measures can actually be computed, to clarify how (or whether) they work. The examples will also help us prove our theorems presented in Section 5 that generalise our observations.

4.1 Example 1

Table 1 shows an example with $M = 3$ classes and $N = 9$ instances,[1] where top probabilities for correct and incorrect cases are shown in blue and red, respectively. Systems Y, Z, W are obtained by *perturbing* (i.e., *hurting*) System X as follows:

[1] Note that, with the exception of (N)BS, the measures discussed in this paper can be applied to situations where $M (\geq 2)$ varies across instances, for example, when the number of answer candidates within a system varies depending on the question: we can still take one probability per instance (e.g., top probability) for the evaluation.

Instance number j		1	2	3	4	5	6	7	8	9
True class		1	1	3	3	2	1	1	3	1
System X	p_j^1	0.4	0.4	0.4	0.2	0.6	0.6	0.8	0.1	0.8
	p_j^2	0.3	0.3	0.3	0.2	0.2	0.2	0.1	0.1	0.1
	p_j^3	0.3	0.3	0.3	0.6	0.2	0.2	0.1	0.8	0.1
System Y	p_j^1	0.4	0.4	0.4	0.2	0.6	0.6	$\underline{0.7}$	0.1	0.8
	p_j^2	0.3	0.3	0.3	0.2	0.2	0.2	$\underline{0.2}$	0.1	0.1
	p_j^3	0.3	0.3	0.3	0.6	0.2	0.2	0.1	0.8	0.1
System Z	p_j^1	0.4	0.4	$\underline{0.6}$	0.2	0.6	0.6	0.8	0.1	0.8
	p_j^2	0.3	0.3	$\underline{0.1}$	0.2	0.2	0.2	0.1	0.1	0.1
	p_j^3	0.3	0.3	0.3	0.6	0.2	0.2	0.1	0.8	0.1
System W	p_j^1	0.4	0.4	$\underline{0.6}$	0.2	0.6	0.6	$\underline{0.7}$	0.1	0.8
	p_j^2	0.3	0.3	$\underline{0.1}$	0.2	0.2	0.2	$\underline{0.2}$	0.1	0.1
	p_j^3	0.3	0.3	0.3	0.6	0.2	0.2	0.1	0.8	0.1

Table 1: First example of perturbing system confidence scores when the system (X) has $M = 3$ answer candidates for each of the $N = 9$ questions given. Top probabilities are shown in blue if correct and in red if incorrect; note that the top-probability-based accuracy is unchanged: $7/9 = 0.778$. The underlines indicate the perturbations.

	X	Y	Z	W
HMR↑	0.557	**0.551**	**0.489**	**0.485**
ECE↓	0.178	**0.189**	$\underline{0.156}$	$\underline{0.167}$
MCE↓	0.267	0.267	$\underline{0.200}$	$\underline{0.233}$
NBS↓	0.130	**0.133**	**0.135**	**0.138**
KS↓	0.178	**0.189**	$\underline{0.156}$	$\underline{0.167}$

Table 2: Summary of results for the first example. Intuitive results are indicated in **bold**; counterintuitive results are indicated by underlines.

Y Pick one top probability that represents a *correct* case, and *lower* it while keeping it the top probability, thereby injecting *underconfidence*;

Z Pick one top probability that represents an *incorrect* case, and *raise* it, thereby injecting *overconfidence*;

W Apply both of the above perturbations.

Note that the above perturbations do not affect the top-probability-based accuracy which is $7/9 = 0.778$ for this example. The perturbed probabilities are underlined in Table 1.

For our task where we are concerned with underconfidence and overconfidence of system responses, we would like to be able to say that Y, Z, and W all *underperform* X. However, for this example, only HMR and NBS satisfy this requirement, as shown in Table 2. Here, the results that we want (intuitive results) are shown in **bold**, and the counterintuitive ones are underlined. Note that HMR is a reward measure (i.e., higher means better), while the others quantify errors (i.e., lower means better), as indicated by the arrows. Below, we demonstrate how some of the numbers in Table 2 are obtained in order to clarify how the measures work (or not). We shall leave the discussion of NBS to Appendix B, in which we provide a different example where NBS gives counterintuitive scores for Y, Z, and W. Recall that, unlike the other measures, NBS relies on the probability for *every* class for each instance.

4.1.1 HMR for Example 1

For System X in Table 1, $O = 0.4 + 0.6 = 1.0, U = 2 * (1 - 0.4) + 2 * (1 - 0.6) + 3 * (1 - 0.8) = 2.6$ (Eq. 5). Since $|I^-| = 2, |I^+| = 7$, $R_O = 1 - 1.0/2 = 0.500$ (Eq. 6) and $R_U = 1 - 2.6/7 = 0.629$ (Eq. 7). Hence X is *more overconfident than underconfident*; note that this observation is not possible with the other measures. Finally, $HMR(X) = 0.557$ (Eq. 8).

Similarly, for System W, $O = 1.2$ (same as Z), and $U = 2.7$ (same as Y); $R_O = 0.400$ (worse than X in terms of overconfidence), and $R_U = 0.614$ (worse than X in terms of underconfidence). Hence $HMR(W) = 0.485$ (worse than X overall).

4.1.2 ECE and MCE for Example 1

The instances in Table 1 are already sorted by top probability and binned for computing ECE (and MCE): we have $B = 3$ bins, each containing three instances. The binwise accuracies (a_b) are $(2/3, 2/3, 3/3)$ for all systems. For X, the average confidences ($\bar{c}_b$) are clearly

	System X					System Z				
j	p_j	cp_j	$GOLD_j$	$cGOLD_j$	δ_j	p_j	cp_j	$GOLD_j$	$cGOLD_j$	δ_j
1	0.4	0.044	1	0.111	0.067	0.4	0.044	1	0.111	0.067
2	0.4	0.089	1	0.222	0.133	0.4	0.089	1	0.222	0.133
3	0.4	0.133	0	0.222	0.089	$\underline{0.6}$	0.156	0	0.222	0.067
4	0.6	0.200	1	0.333	0.133	0.6	0.222	1	0.333	0.111
5	0.6	0.267	0	0.333	0.067	0.6	0.289	0	0.333	0.044
6	0.6	0.333	1	0.444	0.111	0.6	0.356	1	0.444	0.089
7	0.8	0.422	1	0.556	0.133	0.8	0.444	1	0.556	0.111
8	0.8	0.511	1	0.667	0.156	0.8	0.533	1	0.667	0.133
9	0.8	0.600	1	0.778	**0.178**	0.8	0.622	1	0.778	**0.156**

Table 3: Computing the KS scores from the top probabilities shown in Table 1 for X and Z. The δ_j column represents $|cp_j - cGOLD_j|$ in Eq. 4; the maximum δ_j across the rows is the KS score by definition, as indicated in **bold**. Probabilities for correct and incorrect cases are indicated in blue and red, respectively. The underlined probability indicates where System Z differs from X.

$(0.400, 0.600, 0.800)$; on the other hand, for Z which has an overconfidence injected in Bin 1, the average confidences are $(\mathbf{0.467}, 0.600, 0.800)$. Hence, the binwise absolute differences ($|\bar{c}_b - a_b|$ in Eq. 1) are $(0.267, 0.067, 0.200)$ for X, and $(\mathbf{0.200}, 0.067, 0.200)$ for Z. Thus, even though Z is more confident than X about the third instance (and they are both incorrect), $ECE(X) = 0.178, ECE(Z) = 0.156, MCE(X) = 0.267, MCE(Z) = 0.200$. That is, both ECE and MCE say that Z is better.

The above flaw arises as follows. For X, note that $a_1 = 2/3 > \bar{c}_1 = 0.400$: that is, for Bin 1, X is *underconfident on average*. Hence the absolute difference $|\bar{c}_1 - a_1| = 0.267$ actually quantifies how *underconfident* X is for Bin 1. Now, the perturbation introduced in Z *raises* $\bar{c}_1$ (as Z is more confident than X about the third instance), and therefore Z is considered to be "less underconfident" than X for Bin 1. From this discussion, it is clear that binwise averaging of confidences is not a good idea for the purpose of evaluating both overconfidence and underconfidence while trying to separate them, as averaging confounds both phenomena.

Note also that in Table 2, MCE fails to detect the perturbation introduced in Y for Bin 3. This is because, although the average confidence $\bar{c}_3$ is lowered from 0.800 to $(0.7 + 2 * 0.8)/3 = 0.767$ and hence the absolute difference $|\bar{c}_3 - a_3| = |\bar{c}_3 - 1|$ is raised from 0.200 to 0.233, this new value is still smaller than the unchanged absolute difference for Bin 1: $|\bar{c}_1 - a_1| = 0.267$. In other words, when Y is obtained from X by perturbing Bin 3, MCE keeps looking at Bin 1 and ignores the change. Thus, although MCE was proposed to consider extreme cases, binwise averaging of confidences prior to applying the max operator (Eq. 1) can hide what is happening to individual instances.

4.1.3 KS for Example 1

Table 3 shows how KS scores are computed for Systems X and Z shown in Table 1 according to Eq. 4. Note that KS also requires instance sorting, and recall that Table 1 already provides the instances sorted by top probabilities. It can be verified that, even though Z is overconfident about the third instance ($j = 3$) compared to X (where both systems are incorrect), KS says that Z is better.

The above flaw arises as follows. In Table 3, note that $cp_2 = 0.089 < cGOLD_2 = 0.222$ for both systems: the former is much smaller, even though KS requires the cp distribution to align with the $cGOLD$ distribution. In other words, at $j = 2$, the systems are *on the side of underestimation so far*. Therefore, if we *raise* p_3 (from 0.4 to 0.6), this brings the cp distribution "closer" to the $cGOLD$ distribution: it can be verified that, while $cGOLD_3 = 0.222$, $cp_3 = 0.133$ for X and $cp_3 = 0.156$. Hence the counterintuitive result.

4.2 Example 2

In our first example (Tables 1-2), ECE and KS managed to say that Y (perturbed by injecting underconfidence for a correct case) is worse than X. Our second example, shown in Tables 4-5 ($M = 3, N = 9$, with Y, Z, W perturbed as described earlier), show that ECE and KS fail to do so; The same goes for MCE. From Table 4, it can be observed that the top probability of X for the first instance ($j = 1$)

Instance number j		1	2	3	4	5	6	7	8	9
True class		1	1	3	3	2	1	1	3	1
System X	p_j^1	0.5	0.3	0.5	0.2	0.6	0.6	0.7	0.2	0.7
	p_j^2	0.3	0.5	0.3	0.2	0.2	0.2	0.2	0.7	0.2
	p_j^3	0.2	0.2	0.2	0.6	0.2	0.2	0.1	0.1	0.1
System Y	p_j^1	$\underline{0.4}$	0.3	0.5	0.2	0.6	0.6	0.7	0.2	0.7
	p_j^2	0.3	0.5	0.3	0.2	0.2	0.2	0.2	0.7	0.2
	p_j^3	$\underline{0.3}$	0.2	0.2	0.6	0.2	0.2	0.1	0.1	0.1
System Z	p_j^1	0.5	0.3	$\underline{0.6}$	0.2	0.6	0.6	0.7	0.2	0.7
	p_j^2	0.3	0.5	$\underline{0.2}$	0.2	0.2	0.2	0.2	0.7	0.2
	p_j^3	0.2	0.2	0.2	0.6	0.2	0.2	0.1	0.1	0.1
System W	p_j^1	$\underline{0.4}$	0.3	$\underline{0.6}$	0.20	0.6	0.6	0.7	0.2	0.7
	p_j^2	0.3	0.5	$\underline{0.2}$	0.2	0.2	0.2	0.2	0.7	0.2
	p_j^3	$\underline{0.3}$	0.2	0.2	0.6	0.2	0.2	0.1	0.1	0.1

Table 4: Second example of perturbing system confidence scores when the system (X) has $M = 3$ answer candidates for each of the $N = 9$ questions given. Top probabilities are shown in blue if correct and in red if incorrect; note that the top-probability-based accuracy is unchanged: $5/9 = 0.556$. The underlines indicate the perturbations.

	X	Y	Z	W
HMR↑	0.504	**0.498**	**0.486**	**0.480**
ECE↓	0.089	$\underline{0.078}$	**0.100**	0.089
MCE↓	0.167	$\underline{0.133}$	**0.200**	0.167
NBS↓	0.196	**0.201**	**0.198**	**0.204**
KS↓	0.078	$\underline{0.067}$	**0.089**	0.078

Table 5: Summary of results for the second example. Intuitive results are indicated in **bold**; counterintuitive results are indicated by underlines.

has been lowered from 0.5 to 0.4 in order to obtain Y, even though both X and Y are correct for this instance. Below, we examine why ECE, MCE, and KS say that Y is better.

4.3 ECE and MCE for Example 2

From Table 4, the binwise accuracies (a_b) are $(1/3, 2/3, 2/3)$; the average confidences ($\bar{c}_b$) are $(0.500, 0.600, 0.700)$ for X, and $(\mathbf{0.467}, 0.600, 0.700)$ for Y due to the injection of underconfidence. Hence the binwise absolute differences ($|\bar{c}_b - a_b|$) are $(0.167, 0.067, 0.033)$ for X, and $(\mathbf{0.133}, 0.067, 0.700)$ for Y. Therefore, from Eq. 1, MCE (which reflects only Bin 1) and ECE are smaller (i.e., "better") for Y.

The above flaw arises as follows. Note that $a_1 = 1/3 < \bar{c}_1 = 0.500$ for X; hence the absolute difference for Bin 1 actually quantifies *overconfidence*. Therefore, Y, which is less confident in Bin 1 due to the perturbation, is considered to be "less overconfident" than X. Again, it is clear that binwise averaging is not a good idea in our context.

4.4 KS for Example 2

Table 6 shows how KS scores are computed for X and Y shown in Table 4 according to Eq. 4. The left side of the table shows that, for System X, cp_j diverges most from $cGOLD_j$ at $j = 8$ and this is what determines the KS score: $KS(X) = 0.078$. Now, note that $cp_8 = 0.522 > cGOLD_8 = 0.444$: That is, X is *on the side of overestimation* at $j = 8$. Therefore, if we want to reduce the difference between cp_8 and $cGOLD_8$, we could (for example) consider lowering cp_j for every j, by just lowering p_1: this is exactly what the perturbation injected in Y represents. As can be seen in the right side of Table 6, we have "successfully" reduced the difference between cp_8 and $cGOLD_8$: now the maximum difference is observed not only at $j = 8$ but also at $j = 1$ and $j = 5$, and $KS(Y) = 0.067$. Thus, just like ECE and MCE, KS says that Y is better than X, which is counterintuitive.

5 Axioms and Theorems

The examples discussed in Section 4 demonstrated how the measures are actually computed, and how ECE, MCE, and KS can be counterintuitive for our purpose. (As mentioned earlier, counterintuitive cases for NBS are provided in Appendix B.) However, examples are examples: this section clarifies the advantages of HMR in terms of *axioms* that it satisfies, to generalise our previous observations.

	System X					System Y				
j	p_j	cp_j	$GOLD_j$	$cGOLD_j$	δ_j	p_j	cp_j	$GOLD_j$	$cGOLD_j$	δ_j
1	0.5	0.056	1	0.111	0.056	0.4	0.044	1	0.111	**0.067**
2	0.5	0.111	0	0.111	0.000	0.5	0.100	0	0.111	0.011
3	0.5	0.167	0	0.111	0.056	0.5	0.156	0	0.111	0.044
4	0.6	0.233	1	0.222	0.011	0.6	0.222	1	0.222	0.000
5	0.6	0.300	0	0.222	0.078	0.6	0.289	0	0.222	**0.067**
6	0.6	0.367	1	0.333	0.033	0.6	0.356	1	0.333	0.022
7	0.7	0.444	1	0.444	0.000	0.7	0.433	1	0.444	0.011
8	0.7	0.522	0	0.444	**0.078**	0.7	0.511	0	0.444	**0.067**
9	0.7	0.600	1	0.556	0.044	0.7	0.589	1	0.556	0.033

Table 6: Computing the KS scores from the top probabilities shown in Table 4 for X and Z. The δ_j column represents $|cp_j - cGOLD_j|$ in Eq. 4; the maximum δ_j across the rows is the KS score by definition, as indicated in **bold**. Probabilities for correct and incorrect cases are indicated in blue and red, respectively. The underlined probability indicates where System Y differs from X.

5.1 Axioms

All three axioms presented below start with the following common prerequisite. Consider a sequence of binary correctness labels for N instances; the label for Instance i is denoted by $GOLD(i)$. Under this setting, consider System X that returns a sequence $\langle p_1, \ldots, p_N \rangle$ of confidence scores (i.e., probabilities) for the same N instances, where the scores have been sorted in ascending order (just for computing ECE, MCE, and KS). Let i_j denote the j-th instance in the sorted list; then the corresponding sequence of the correctness labels can be denoted as $\langle GOLD(i_1), \ldots, GOLD(i_N) \rangle$.

Axiom 5.1 (Axiom-U) *Consider System Y, obtained by perturbing the confidence score sequence of System X as follows. Suppose that for one particular instance i_j s.t. $GOLD(i_j) = 1$ (i.e., X is correct about the j-th instance), we managed to replace p_j with $q_j (< p_j)$ without affecting the prediction outcome (i.e., Y is still correct about this instance). Since the confidence is now **lower** for this **correct** case and nothing else has changed, Y should not be considered superior to X.*

Axiom 5.2 (Axiom-O) *Consider System Z, obtained by perturbing the confidence score sequence of System X as follows. Suppose that for one particular instance $i_{j'}$ s.t. $GOLD(i_{j'}) = 0$ (i.e., X is incorrect about the j'-th instance), we managed to replace $p_{j'}$ with $q_{j'} (> p_{j'})$ without affecting the prediction outcome (i.e., Z is still incorrect about this instance). Since the confidence is now **higher** for this **incorrect** case and nothing else has changed, Z should not be considered superior to X.*

	Axiom-U (X→Y)	**Axiom-O** (X→Z)	**Axiom-UO** (X→W)
HMR↑	YES	YES	YES
ECE↓	NO	NO	NO
MCE↓	NO	NO	NO
NBS↓	NO	NO	NO
KS↓	NO	NO	NO

Table 7: Summary of whether each measure satisfies the three axioms or not.

Axiom 5.3 (Axiom-UO) *Consider System W, obtained by applying both of the perturbations mentioned above. Compared to X, the confidence for the **correct** case is **lower** and the confidence for the **incorrect** case is **higher** and nothing else has changed. In this situation, W should be considered strictly inferior to X.*

Note that **Axiom-U** (**Axiom-O**) tolerates evaluation measures that cannot tell the difference between X and Y (X and Z); on the other hand, **Axiom-UO** requires measures to say that W is strictly worse than X.

Table 7 provides a summary of whether each measure satisfies the three axioms or not. Below, we provide the proofs.

5.2 HMR Satisfies All Tree Axioms

Theorem 5.1 (U-HMR) *HMR satisfies **Axiom-U**.*

Proof: The perturbation described in **Axiom-U** does not affect O (Eq. 5) and hence does not affect R_O either (Eq. 6): for brevity, let $c = R_O$ denote the unaffected reward. On the other hand, the perturbation *increases* U (Eq. 5) and hence *decreases* R_U (Eq. 7): that is, if we let a and b denote the R_U for X and the R_U for Y, respectively, then $a > b (\geq 0)$. From Eq. 8, $\mathit{HMR}(X) = 2ca/(c+a)$ since $a > 0$. We need to show that $\Delta = \mathit{HMR}(X) - \mathit{HMR}(Y) \geq 0$.

Suppose that $c = 0$, i.e., $O = |I^-|$ (Eq. 6), that is, **both X and Y are 100% confident for every incorrect case**. Then $\mathit{HMR}(X) = 0/a = 0$. If $b > 0$, $\mathit{HMR}(Y) = 2cb/(c+b) = 0/b = 0$; if $b = 0$, then $c = b = 0$ so $\mathit{HMR}(Y) = 0$ (Eq. 8); Either way, $\Delta = 0 - 0 = 0$.

Otherwise (i.e., if $c > 0$), $\Delta = 2ca/(c+a) - 2cb/(c+b) = 2c^2(a-b)/(c+a)(c+b) > 0$.

Theorem 5.2 (O-HMR) *HMR satisfies* ***Axiom-O****.*

Proof: The perturbation described in **Axiom-O** does not affect U (Eq. 5) and hence does not affect R_U either (Eq. 7): for brevity, let $c = R_U$ denote the unaffected reward. On the other hand, the perturbation *increases* O (Eq. 5) and hence *decreases* R_O (Eq. 6): that is, if we let a and b denote the R_O for X and the R_O for Z, respectively, then $a > b (\geq 0)$. Since $a > 0$, $\mathit{HMR}(X) = 2ac/(a+c)$. We need to show that $\Delta' = \mathit{HMR}(X) - \mathit{HMR}(Z) \geq 0$.

Suppose that $c = 0$, i.e., $U = |I^+|$ (Eq. 7), that is, **both X and Z are 0% confident for every correct case**. Then $\mathit{HMR}(X) = 0/a = 0$. If $b > 0$, $\mathit{HMR}(Z) = 2bc/(b+c) = 0/b = 0$; if $b = 0$ as well, then $b = c = 0$ so $\mathit{HMR}(Z) = 0$ (Eq. 8). Either way, $\Delta' = 0 - 0 = 0$.

Otherwise (i.e., if $c > 0$), $\Delta' = 2ac/(a+c) - 2bc/(b+c) = 2c^2(a-b)/(a+c)(b+c) > 0$.

Theorem 5.3 (UO-HMR) *HMR satisfies* ***Axiom-UO****.*

Proof: Based on the proofs of **Theorems U-HMR** and **O-HMR**, it is clear that the two perturbations described in **Axiom-UO** decrease both R_U (due to the j-th instance) and R_O (due to the j'-th instance). Hence the harmonic mean (Eq. 8) also decreases; that is, $\mathit{HMR}(X) - \mathit{HMR}(W) \geq 0$. Moreover, from the proofs of U-HMR and O-HMR, it follows that the equality can hold only when both X and W are 100% confident for every incorrect case and 0% confident for every correct case. However, we know that this is not possible: if X is 100% confident for every incorrect case, it is not possible to further inject overconfidence; if X is 0% confident for every correct case, it is not possible to further inject underconfidence. Hence, $\mathit{HMR}(X) > \mathit{HMR}(W)$ holds: W is strictly inferior to X.

	Axiom-U	**Axiom-O**	**Axiom-UO**
	(X→Y)	(X→Z)	(X→W)
ECE ↓	Example 2	Example 1	Example 1
MCE↓	Example 2	Example 1	Example 1
NBS↓	Example 3	Example 3	Example 3
KS↓	Example 2	Example 1	Example 1

Table 8: Counterexamples that show that these measures do not satisfy the axioms. Examples 1 and 2 are given in Tables 1 and 4, respectively. Example 3 is provided in Appendix B as NBS relies on the probability for every class unlike the other measures.

5.3 ECE, MCE, NBS, and KS Satisfy None of the Axioms

To prove that none of ECE, MCE, NBS, and KS satisfy any of the axioms, providing one actual counterexample for each situation suffices. Table 8 provides the counterexamples necessary.

6 Conclusions and Future Work

For the purpose of penalising both overconfidence and underconfidence in system responses while balancing the two, we proposed a simple and intuitive evaluation measure called HMR. We proved that HMR satisfies our axioms (i.e., penalising a lowered confidence for a correct response, penalising a raised confidence for an incorrect response, and penalising a system that reflects both perturbations), and that existing calibration measures do not. Hence, while we do not claim that HMR should replace existing calibration measures in all calibration tasks, we do recommend its use in tasks where our axioms make sense.

We designed HMR primarily for conversational search systems where each response is either *correct* or not and has a confidence score; the score could represent a top probability (or more generally, the n-th highest probability) among the probabilities for M different response candidates; the candiates may be generated by the system itself or given to the system from outside, as in multiple choice questions. However, HMR can be used in any task where the system response has a binary gold label and one confidence score. For example, the gold label could represent whether the system response is *harmful* or not in a *content moderation* task. We plan to explore more ways to utilise HMR in future work, for example, in a shared conversational search task.

Limitations

Figure 1 was obtained through a pilot experiment with Google Bard (as of November 2023) to exemplify a situation where the same system can be overconfident and underconfident depending on test instances. The experiment does not represent an actual case where a system has a confidence score for each of its own responses; instead, Google Bard returned a confidence score to each response that was fed to it. Evaluating real situations with HMR is left for future work. However, note that our axioms and theorems do not depend on data.

We designed HMR specifically for tasks where overconfidence and underconfidence should be considered separately, and balancing between the two is important. We do not claim that all calibration measures should be replaced by ours in traditional calibration tasks; we only claim that HMR should be preferred over existing calibration measures for tasks where our axioms make sense.

Ethics Statement

The present study was ethically motivated. We started this work because the recent LLM-based conversational systems often hallucinate, and not only the fluency but also the confidence of their responses tend to fool the users. The misinformation may negatively impact the users, the downstream tasks, and society at large. The information thus spread may then be fed to LLMs as training data. We believe that it is the responsibility of NLP researchers to prevent such vicious circles. Suppressing both overconfidence and underconfidence means that the system will be just as modest about the credibility of their responses as it should be. Our hope is that this work will help the research community advance towards that goal.

References

Glenn W. Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks.

Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. 2021. Calibration of neural networks using splines.

William L. Hays. 1994. *Statistics (Fifth Edition)*. Harcourt Brace College Publishers.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Ananya Kumar, Percy Liang, and Tengyu Ma. 2020. Verified uncertainty calibration.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: a survey and guideline for evaluating large language models' alignment.

Khanh Nguyen and Brendan O'Connor. 2015. Posterior calibration and exploratory analysis for natural language processing models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1587–1598, Lisbon, Portugal. Association for Computational Linguistics.

Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. 2020. Measuring calibration in deep learning.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Gwenyth Portillo Wightman, Alexandra Delucia, and Mark Dredze. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 326–362, Toronto, Canada. Association for Computational Linguistics.

Tetsuya Sakai. 2021. Evaluating evaluation measures for ordinal classification and ordinal quantification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2759–2769, Online. Association for Computational Linguistics.

Weng Tam, Xiao Liu, Kaixuan Ji, Lilong Xue, Jiahua Liu, Tao Li, Yuxiao Dong, and Jie Tang. 2023. Parameter-efficient prompt tuning makes generalized and calibrated neural text retrievers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13117–13130, Singapore. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. 2019. Evaluating model calibration in classification. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 3459–3467. PMLR.

Haotian Wang, Zhen Zhang, Mengting Hu, Qichao Wang, Liang Chen, Yatao Bian, and Bingzhe Wu. 2023. RECAL: Sample-relation guided confidence calibration over tabular data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7246–7257, Singapore. Association for Computational Linguistics.

Polina Zablotskaia, Du Phan, Joshua Maynez, Shashi Narayan, Jie Ren, and Jeremiah Liu. 2023. On uncertainty calibration and selective generation in probabilistic neural summarization: A benchmark study. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2980–2992, Singapore. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference.

Chiwei Zhu, Benfeng Xu, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2023. On the calibration of large language models and alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9778–9795, Singapore. Association for Computational Linguistics.

A A Pilot Experiment with SWAG

This section supplements Section 1 by describing exactly how we obtained Figure 1.

We first downloaded the raw SWAG validation dataset (Zellers et al., 2018): `https://github.com/rowanz/swagaf/blob/master/data/val_full.csv`. We chose this data set just to illustrate our point that underconfidence can sometimes be a problem along with the more widely known overconfidence problem of system responses. The dataset contains 20,006 records, where each record represents a *video caption completion task* instance, consisting of a *startphrase* (i.e., an incomplete caption), a *gold* response (i.e., the correct final phrase that completes the caption), and several *distractors* (i.e., phrases that are considered less good than the gold response or even completely out of context). From this multiple choice data set, we randomly extracted 100 records where each record consists of a startphrase, the gold response, and exactly one distractor that was labelled "unl(ikely)" (i.e., the distractor sounds ridiculous or impossible given the context). Based on this sample, we created 100 *gold response prompts* as well as 100 *distractor response prompts*: Figure 2 provides an example gold response prompt, together with the corresponding distractor response prompt. Note that the only difference between the two is the content of "YOUR FINAL PHRASE" part.

We randomly shuffled the 200 prompts thus created as it is known that conversational search responses may be affected by previous prompts: they "remember" to some extent. That is, a gold response prompt and its distractor counterpart were treated as two independent records during the random shuffling so that the system cannot directly compare the gold and distractor final phrases. The prompts were fed to Google Bard one by one on November 15, 2023; when the 108th prompt was entered, Google Bard refused to continue the work; hence the work was resumed on November 21, 2023 and was completed on the same day. In this way, we obtained a confidence score in the 0-1 range for each of the 200 prompts (with 4 exceptions as described later). For example, for the prompts shown in Figure 2, Google Bard returned 0.2 (representing underconfidence, as the final phrase is gold) and 0.6 (representing overconfidence, as the final phrase is a distractor), respectively. This record is highlighted with a balloon in Figure 1.

Despite our explicit instruction in the template part of the prompts for Google Bard to return a confidence score only, Google Bard occasionally returned some additional information such as YouTube links or sentences. For these cases, we recorded the confidence scores along with "redundant" as a comment. Also, Google Bard failed to return a confidence score for 4 of the 200 prompts; for these cases, we recorded "NA" instead of a confidence score. (We actually tried both Google Bard and the New Bing for this experiment but the latter flatly refused to return confidence scores from the outset, as of November 15, 2023.)

To ensure that Figure 1 can be reproduced (despite the lack repeatability of our Google Bard-based experiment itself as its behaviour keeps changing), we will make the 200 shuffled prompts and the confidence scores returned by Google Bard publicly available upon paper acceptance. (The data is shared with the ACL ARR reviewers as an accompanying zip file.)

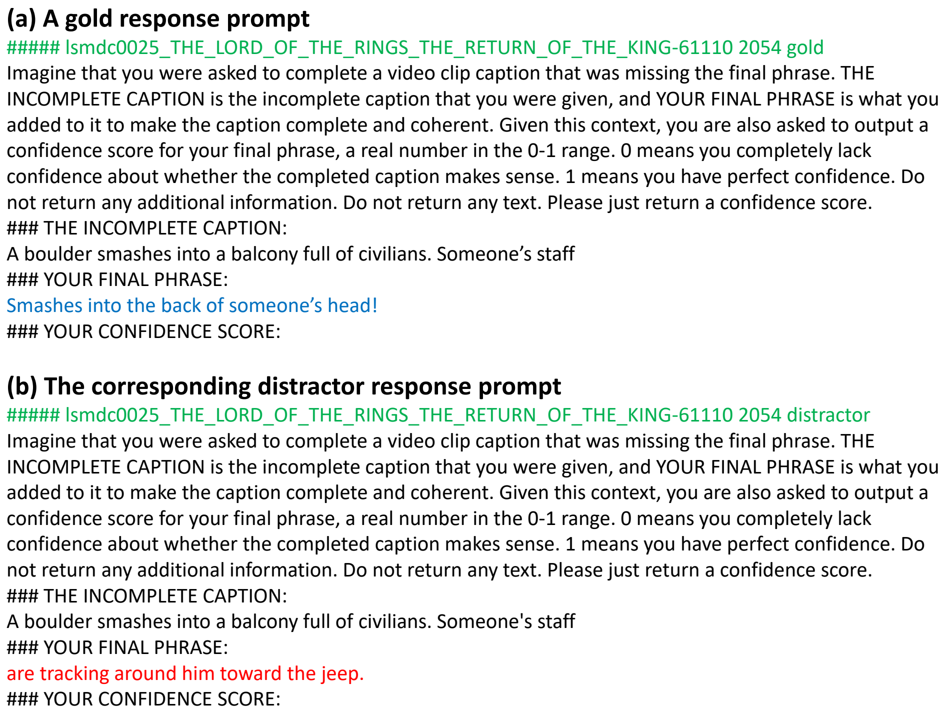
B Counterexamples for NBS

This section discusses our third example, which demonstrates that NBS can be counterintuitive when the perturbations described in Section 4.1 are applied to System X in order to obtain Y, Z, and W.

Table 9 presents our third example with $M = 6, N = 3$; Table 10 shows the HMR, NBS, and KS scores computed from Table 9. ECE and MCE are omitted here, as these measures require instance binning and binwise averaging of confidences but we only have three instances.

For X, the sum of squared errors (Eq. 2) for the third instance ($j = 3$) is $(0.6 - 1)^2 + 0.4^2 = 0.32$. In contrast, for Y, the corresponding value is $(0.5 - 1)^2 + 5 * 0.1^2 = 0.30$; this is why Y is considered better than X. Meanwhile, for X, the sum of squared errors for the second instance ($j = 2$) is $0.4^2 + 0.3^2 + 3 * 0.1^2 = 0.68$. In contrast, for Z, the corresponding value is $0.5^2 + 0.4^2 + 0.1^2 = 0.62$; this is why Z is considered better than X. Finally, W is also considered better than X according to NBS, as W reflects both of the above changes in sum of squared errors.

As a final remark, note that KS completely fails to detect the perturbations in Table 10.



(a) A gold response prompt

##### lsmdc0025_THE_LORD_OF_THE_RINGS_THE_RETURN_OF_THE_KING-61110 2054 gold

Imagine that you were asked to complete a video clip caption that was missing the final phrase. THE INCOMPLETE CAPTION is the incomplete caption that you were given, and YOUR FINAL PHRASE is what you added to it to make the caption complete and coherent. Given this context, you are also asked to output a confidence score for your final phrase, a real number in the 0-1 range. 0 means you completely lack confidence about whether the completed caption makes sense. 1 means you have perfect confidence. Do not return any additional information. Do not return any text. Please just return a confidence score.

### THE INCOMPLETE CAPTION:

A boulder smashes into a balcony full of civilians. Someone’s staff

### YOUR FINAL PHRASE:

Smashes into the back of someone’s head!

### YOUR CONFIDENCE SCORE:

(b) The corresponding distractor response prompt

##### lsmdc0025_THE_LORD_OF_THE_RINGS_THE_RETURN_OF_THE_KING-61110 2054 distractor

Imagine that you were asked to complete a video clip caption that was missing the final phrase. THE INCOMPLETE CAPTION is the incomplete caption that you were given, and YOUR FINAL PHRASE is what you added to it to make the caption complete and coherent. Given this context, you are also asked to output a confidence score for your final phrase, a real number in the 0-1 range. 0 means you completely lack confidence about whether the completed caption makes sense. 1 means you have perfect confidence. Do not return any additional information. Do not return any text. Please just return a confidence score.

### THE INCOMPLETE CAPTION:

A boulder smashes into a balcony full of civilians. Someone's staff

### YOUR FINAL PHRASE:

are tracking around him toward the jeep.

### YOUR CONFIDENCE SCORE:

Figure 2: An example of a gold response prompt and the corresponding distractor response prompt. The green parts are comments, not part of the prompts.

Instance number j		1	2	3	True class		3	2	1
System X	p_j^1	0.3	0.4	0.6	System Y	p_j^1	0.3	0.4	$\underline{0.5}$
	p_j^2	0.4	0.3	0.4		p_j^2	0.4	0.3	$\underline{0.1}$
	p_j^3	0.1	0.1	0		p_j^3	0.1	0.1	$\underline{0.1}$
	p_j^4	0.1	0.1	0		p_j^4	0.1	0.1	$\underline{0.1}$
	p_j^5	0.1	0.1	0		p_j^5	0.1	0.1	$\underline{0.1}$
	p_j^6	0	0	0		p_j^6	0	0	$\underline{0.1}$
System Z	p_j^1	0.3	$\underline{0.5}$	0.6	System W	p_j^1	0.3	$\underline{0.5}$	$\underline{0.5}$
	p_j^2	0.4	$\underline{0.4}$	0.4		p_j^2	0.4	$\underline{0.4}$	$\underline{0.1}$
	p_j^3	0.1	0.1	0		p_j^3	0.1	0.1	$\underline{0.1}$
	p_j^4	0.1	$\underline{0}$	0		p_j^4	0.1	$\underline{0}$	$\underline{0.1}$
	p_j^5	0.1	$\underline{0}$	0		p_j^5	0.1	$\underline{0}$	$\underline{0.1}$
	p_j^6	0	0	0		p_j^6	0	0	$\underline{0.1}$

Table 9: Third example of perturbing system confidence scores when the system (X) has $M = 6$ answer candidates for each of the $N = 3$ questions given. Top probabilities are shown in blue if correct and in red if incorrect; note that the top-probability-based accuracy is unchanged: $1/3 = 0.333$. The underlines indicate the perturbations.

	X	Y	Z	W
HMR↑	0.600	**0.545**	**0.574**	**0.524**
NBS↓	0.116	$\underline{0.114}$	$\underline{0.112}$	$\underline{0.111}$
KS↓	0.200	0.200	0.200	0.200

Table 10: Summary of results for the third example. Intuitive results are indicated in **bold**; counterintuitive results are indicated by underlines. As there are only three instances, ECE and MCE (which require instance binning) are omitted.