Evaluating the Impact of Explicit Hate Speech Definitions on the Stability of LLM-based Hate Speech Classification

Rodrigo Santos

Supervisor(s): Pradeep Murakannaiah, Urja Khurana

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Rodrigo Santos
Final project course: CSE3000 Research Project
Thesis committee: Pradeep Murakannaiah, Urja Khurana, Cynthia Liem

Abstract

Automated hate speech detection is crucial to keep up with the high demand for moderation online, yet current models struggle to produce stable and consistent results. While metrics such as accuracy evaluate a model's overall performance, they fail to detect instability, meaning predictions on identical inputs fluctuate. Better metrics exist that can detect this, such as micro-consistency, which looks at the consistency on the individual test case level. This paper looks at what the impact is of providing explicit definitions of hate speech to LLMs for hate speech classification, using micro-consistency metrics and uncertainty metrics. The research was done using the Llama-3-8B-Instruct model for binary classification on the HateCheck dataset. The results show that providing explicit definitions for hate speech classification using zero-shot prompting worsened micro-consistency and uncertainty, and that the differences are statistically significant. However, more research is required to conclude with certainty that this decline in stability is caused by the model's inherent limitations, rather than a suboptimal setup for this task.

1 Introduction

As the internet has grown significantly over the last few decades, the task of moderating has become more complex. Due to the growing scale, it is infeasible to rely solely on human moderators. To fix this issue, there are algorithms that can detect unwanted behavior online automatically. An example of such behavior is hate speech, which is generally defined as communication that attacks protected characteristics such as race, religion or sexual orientation. This is a difficult problem to solve, because hate speech is context dependent. Thus proper detection goes beyond banning specific words. There are also different definitions of hate speech, and different levels of tolerance that can be chosen depending on the needs of a system. To ensure the algorithm is effective, it is important for these automated moderation systems to be both fair and reliable. Therefore, it is useful to research different methods for improving the stability of these algorithms. Here stability is used as an umbrella term to refer to the micro-consistency and uncertainty.

The current findings by [10] show that metrics such as accuracy only evaluate overall performance, but ignore weaknesses on the level of individual hate speech categories or test cases. The consistency metric (CON) by [2] calculates how well a model can reproduce an evaluation when rerunning the LLM on the same dataset. They also made a distinction between performance on the macro level which is measured by accuracy, and performance on the micro level which is measured by the consistency metric. We will refer to these concepts as macro-consistency and micro-consistency respectively, while consistency refers to the metric.

[4] stated that modern neural networks are more overconfident than older models, meaning that the gap between the accuracy of the predictions and the confidence of the models in their predictions has increased. This gap is quantified using the Expected Calibration Error (ECE) score, which is a measure of calibration originally proposed by [8]. [4] demonstrates that while the accuracy of the predictions has increased throughout the years, calibration has become worse. There is a trade-off between accuracy and calibration.

Finally, uncertainty is a measure of the model's knowledge. [5] defined two different types of uncertainty: aleatoric uncertainty is caused by inherent randomness of the dataset, and epistemic uncertainty is caused by the model's limited information. For the purpose of analyzing stability, epistemic uncertainty is the main focus, since it isolates the model's capabilities and excludes randomness. This is important because instead of taking micro-consistency at face value, uncertainty provides a way to measure what the model actually knows.

The knowledge gap is what effect providing explicit hate speech definitions to the LLM has on the stability of the model. Thus, the main research question discussed in this paper is the following: **"Does providing a definition of hate speech to an LLM improve the micro-consistency and uncertainty on hate speech classification?".** The following subquestions were derived:

- **SQ1**: What are the best metrics to quantify micro-consistency and uncertainty in the context of hate speech classification?
- **SQ2**: Which configuration parameters (such as choice of LLM, hyperparameters and prompts) help improve model calibration?
- **SQ3**: How do explicit hate speech definitions impact micro-consistency and uncertainty compared to the baseline without explicit definition?

The main contribution of this research is an evaluation of the Llama-3-8B-Instruct model on the HateCheck dataset. The baseline performance is compared to the performance with explicit hate speech definitions. The results are then analyzed to determine whether the definition meaningfully improved the micro-consistency and uncertainty, and whether this is statistically significant.

2 Related Work

2.1 Model calibration

The main way to tweak the responses of LLMs is using the temperature hyperparameter. This value allows the creativity/randomness of LLM responses to be configured, and is thus important for the calibration of LLMs. Multiple methods for calibrating the models were evaluated by [4]. The conclusion was that using temperature scaling is the simplest and most effective method. Temperature scaling divides all the raw logits by a constant temperature value, which brings the average confidence level down to correct for overconfidence. [6] showed that modern LLMs are overconfident due to the Reinforcement Learning from Human Feedback (RLHF) policy. This is a policy used in machine learning to train models, which prioritizes aligning AI responses to human values over proper calibration. In this paper they corrected for the overconfidence using a temperature of 2.5.

For our research, that same overconfidence was found for the logits, and temperature scaling was used to fix this issue. While the $T = 2.5$ value helped calibrate the model in the paper by [6], using such high temperatures was found to distort the output probabilities in our research.

2.2 Model configuration

In the paper by [2], the method used for generating multiple samples is non-deterministic. The same dataset is evaluated by the LLM multiple times with different seeds. This paper also talks about how they used LoRA (Low-Rank Adaptation) to fine-tune the LLMs. The advantage of using LoRA is that you can train the LLM's neural network for a specific task, instead of passing all task information through prompts.

For our research, the sampling was done from the output probability distribution instead of rerunning the LLM. This is because LLM seeds do not affect the logit values produced by the model. In addition, system prompts were used instead of LoRA to test the unmodified performance of Llama-3-8B-Instruct.

Greedy decoding is a strategy where the LLM selects the token with the highest probability when generating the response. [14] found that greedy decoding does not equal determinism, which is important when rerunning samples. This paper also discusses various methods to achieve determinism with greedy decoding, such as using FP32 precision (single-precision floating-point, or float32). They found that this improves reproducibility, due to the higher precision.

Despite this, BF16 (Brain Floating Point 16) was used over FP32 in our research. BF16 is a common data type for running LLMs, due to requiring half the memory of FP32. This also allows faster computation, which was useful given time constraints. To achieve determinism with greedy decoding using BF16, batching was used. Batching refers to grouping multiple prompts into a single request to the LLM. This fixed the non-determinism with greedy decoding issue.

2.3 Micro-Consistency and Uncertainty in LLMs

For micro-consistency, the core metric used is the consistency (CON) metric by [2]. This paper discusses scoring functions to compare individual test case classifications. For simplicity they use a binary classification function, which is also well-suited for the task of hate speech classification.

The consistency metric by [2] was adapted in our research to allow for comparison between more than two models simultaneously. From this, the mean consistency (mCON) metric was derived. This allows comparison between an arbitrary number of models $k \geq 2$.

For uncertainty, many different metrics exist. The main ones are mentioned by [12], which include predictive entropy (PE) and mutual information (MI). Predictive entropy measures the total uncertainty, while mutual information measures specifically the epistemic uncertainty. Another commonly used metric is the Brier score (BS). Finally, more sophisticated metrics exist such as semantic entropy (SE) by [7], which takes into account the meaning of the model's responses.

For our research, the predictive entropy and Brier score metrics were chosen. This is because these are common and reliable metrics for uncertainty and calibration. Mutual information was excluded due to being more computationally expensive, and semantic entropy was excluded because it is not suitable for a binary classification task.

3 Methodology

The core task is to perform binary classification on the HateCheck dataset to determine whether each test case is hate speech or not. This is then repeated for different definitions of hate speech provided to the LLM, including a baseline which does not include an explicit definition. The classifications and logits are then used to quantify micro-consistency and uncertainty, so the impact of the explicit definitions can be analyzed.

3.1 Dataset and Model

The model used to run the experiments is Llama-3-8B-Instruct, a decoder-only large language model (LLM) using the transformer architecture, which was developed by Meta. This model was chosen because it is a standard choice in the field of NLP research, since 8B parameters provide an effective balance between execution speed and response quality. By using the Instruct version, the LLM will follow instructions more precisely. This is useful for both explaining the task of binary classification and for providing the hate speech definitions.

The dataset used for these experiments is the HateCheck dataset by [10], which provides examples of both hateful speech and non-hateful speech. The dataset consists of 3728 different test cases. The main information used is the test cases and the (binary) labels. While this dataset also contains functionalities, which give more information about what type of speech each test case is, they are outside the scope of this research. This dataset contains 29 different functionalities to ensure the test cases are diverse, which makes the dataset as a whole more representative of real online hate speech.

3.2 Background

Temperature Scaling

For each definition there is a shared temperature scaling factor that is used, which also applies to derived samples. The optimal temperature for a definition is found by minimizing the negative log-likelihood (NLL). This forces the model to be more accurate with the logits, by punishing overconfidence. To turn the calibrated logits into usable probabilities, the softmax activation function is used. This is crucial to calculate the metrics that rely on probabilities instead of logit values.

Determinism and Sampling

For the experiments the LLM was run deterministically, which is also known as greedy decoding. Here we create static logits of the entire dataset once per definition. This is because the seed of an LLM does not affect the logits, which makes naive rerunning redundant and computationally expensive. Instead, post-hoc stochastic label sampling was used. The idea is creating different samples from the logits, despite only having run the LLM itself once per definition. This way

we split the process into retrieving the logits, and sampling over the logits post-hoc.

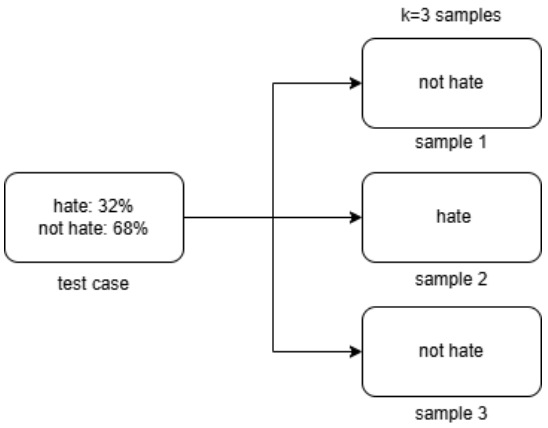

Figure 1: Stochastic Label Sampling example using $k = 3$ flowchart

A flowchart with an example of stochastic label sampling is available in Figure 1. This works by creating k different samples per test case, where k different seeds are picked. These seeds remain consistent across definitions to ensure fairness. For each seed, we reevaluate the classifications non-deterministically based on the calculated probabilities. Once all the test cases are recalculated k times, we have k different samples per test case. This is much faster than prompting the LLM k times.

Metrics

To quantify micro-consistency and uncertainty, the following metrics were used:

- CON (consistency): The consistency metric was defined by [2] as follows:

$$CON(A, B) = \frac{1}{N} \sum_{t=1}^{N} \pi_{A,B}(t) \tag{1}$$

  Here A and B are two (distinct) models, N is the number of test cases, and $\pi_{A,B}(t)$ is a binary scoring function that returns 0 if the models disagree on test case t, and 1 if they agree. This gives the ratio of identical predictions when evaluating two different models on the same dataset. The value provides information about the degree of agreement between the models. Note that in our research the models are going to be distinct samples of the same model.

- mCON (multiple Consistency): This metric was derived from the consistency metric using the following formula:

$$mCON(S) = \frac{1}{\binom{|S|}{2}} \sum_{i<j} CON(S_i, S_j) \tag{2}$$

  Here CON refers to the consistency metric by [2] shown above, S is the set of all samples belonging to one definition, S_i and S_j represent two distinct samples in that set. Due to the limitation of only being able to compare two different samples in the original consistency metric, it was extended to work for comparison between any number $k \geq 2$ samples. This is useful because this allows us to compare any number $k \geq 2$ samples per test case, which is required for proper analysis of micro-consistency. It works by taking the average CON of all possible pairings of samples. Higher values demonstrate resilience towards changing the classification across reruns, which corresponds to better micro-consistency.

- PE (Predictive Entropy): The following formula is used to calculate this metric:

$$PE(\mathbf{p}) = -\sum_{c=1}^{C} p_c \log(p_c) \tag{3}$$

  Here $\mathbf{p}$ is the probability vector, C is the total number of classes ($C = 2$ for binary classification), and p_c is the probability for a given class c. This metric is used to quantify the total uncertainty of both the model and the dataset, which consists of the aleatoric uncertainty and the epistemic uncertainty. Lower values indicate less entropy, meaning there is less uncertainty.

- ECE (Expected Calibration Error): ECE is calculated using this formula:

$$ECE(\mathbf{p}) = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \tag{4}$$

  Here M is the number of bins ($M = 10$), B is set of all M bins, and B_m is the mth bin. Finally, the acc and $conf$ functions calculate the accuracy and mean confidence respectively, for a given bin. Here the 10 bins are equally spaced for the range from 0.0 to 1.0. ECE calculates how well calibrated a model is, by using the difference between the accuracy and the mean confidence. Lower values indicate that the confidence matches the accuracy more closely, which corresponds to more effective calibration.

- BS (Brier Score): Brier score is calculated using the following formula:

$$BS(\mathbf{p}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} (p_i - y_i)^2 \tag{5}$$

  Here $\mathbf{y}$ is the ground truth vector (labels) for the dataset, N is the total number of test cases ($N = 3728$), and y_i and p_i are the actual label and predicted probability for test case i respectively. This metric quantifies both the accuracy of the probabilistic predictions and the calibration of a model. Since the Brier score acts as a MSE (Mean Squared Error) for probabilities, lower values indicate smaller errors and better calibration.

- CR (Consensus Ratio): The consensus Ratio is calculated as follows:

$$CR = \frac{1}{kN} \sum_{i=1}^{N} \max(c_{i,\text{hateful}}, c_{i,\text{non-hateful}}) \tag{6}$$

Here k is the number of samples per test case ($k = 50$), and $c_{i,\text{hateful}}$ and $c_{i,\text{non-hateful}}$ are the number of samples that predicted each outcome. This metric is used to quantify micro-consistency by considering the majority vote per test case. The values range from 0.5 for total disagreement, and 1.0 for perfect agreement. Higher values show that the model is more consistently able to replicate classifications on reruns.

3.3 Experiments

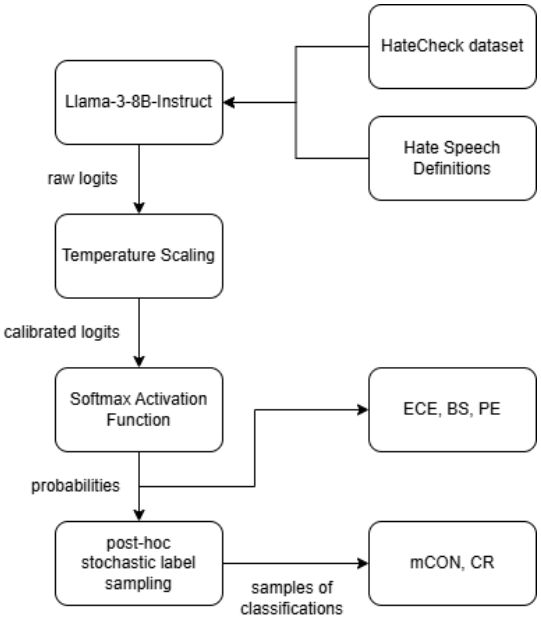

Figure 2: Experiments flowchart

A flowchart with the core of the experiments is available in Figure 2. The Llama-3-8B-Instruct model was tasked with performing binary classification (see prompt in Appendix C.1) on the entirety of the HateCheck dataset, with 3 different definitions of hate speech (see definitions in Appendix B):

1. Baseline: Uses no explicit definition, relying on the default behavior instead.
2. Bulgarian legal definition: Uses 2 articles from Bulgarian law to define hate speech.
3. Reddit rule 1: Reddit's main rule regarding speech on the platform.

All classifications were performed using zero-shot prompting, meaning no examples were provided to the LLM, just the definitions and the task. For each prompt evaluated by the LLM, the logits for hate and not-hate classifications were collected. Subsequently, the optimal temperature was calculated on the unscaled logits separately for each definition. Then all the logits were scaled using the optimal temperature to improve the model calibration. To turn these calibrated logits into usable probabilities to calculate the metrics, the softmax activation function was applied to the calibrated logits. This was then used to calculate PE, ECE and Brier score.

To calculate mCON and CR, multiple samples were created from the static logits. To do this, post-hoc stochastic label sampling was used. Analysis showed that the metrics converge when increasing the number of samples k, and that $k = 50$ was close to the limit. Thus, this number was chosen as the standard across all results, to ensure high accuracy and efficient calculations. Using these results, the different metrics were visualized in tables, graphs, violin plots, and heatmaps, to analyze the effect it had on micro-consistency and uncertainty. Finally, statistical significance was tested, and Spearman's rank correlation [11] was calculated for the metrics. Spearman's rank correlation is used to quantify the correlation between micro-consistency and uncertainty.

4 Experimental Setup

4.1 Batching and Inference Strategy

Instead of providing the test cases to the LLM one by one, they were passed in batches of 32.

For batching, all prompts have to be the same size. Left padding was used to achieve this, which means prepending the string with spaces. This is a good strategy because Llama-3-8B-Instruct is a decoder-only LLM, where the position of the data matters. By putting the padding on the left, the actual instructions and data itself are the last things the LLM sees, which improves the quality of the responses.

To explain the hate speech classification task to the LLM and to provide the explicit definitions of hate speech, system prompts were used. This is the standard way to provide constraints and instructions to the LLM. For the binary classification, the LLM was tasked to respond with '1' if it thought the test case was hate speech, and '0' if not. To force a binary classification, the response of the LLM was limited to a single token. The test cases from the HateCheck dataset were provided to the LLM using user prompts.

4.2 Sanity Check

One thing to consider is the possibility of a refusal to evaluate the test cases, which is common in modern LLMs. However, since the LLM response was limited to a single token, there is a risk of a silent refusal, where a classification is chosen arbitrarily. To confirm this was not the case, the entire HateCheck dataset was evaluated by the LLM on the baseline definition with a modified prompt (see prompt in Appendix C.2) that explicitly states a third classification option for refusal. Out of all 3728 test cases, no test case was refused. This confirmed that the model was not silently refusing any test cases. Finally, the original prompt was substituted back in to avoid adding unnecessary noise to the prompt.

5 Results

5.1 Temperature Scaling

Table 1 shows that the optimal temperatures for each definition far exceed the typical temperature scaling range. According to [3], this range is typically 0.0 to 1.5, but may go up to 2.0. This is useful for more experimental purposes, such as analyzing stability. The high optimal temperatures show that,

Hate Speech Definition	Optimal Temperature (T)
Baseline	3.145
Bulgarian Law	4.345
Reddit Definition	5.012

Table 1: Optimal temperature scaling factors per definition.

regardless of definition, the probabilistic predictions are extremely overconfident pre-calibration.

While it is not strictly necessary to apply the optimal temperature for the minimized negative log-likelihood, optimal values this high show that there is a deeper issue with the pre-calibrated logits that temperature scaling is unable to fix. The explicit definitions have the highest optimal temperatures, meaning the calibration became worse after adding explicit definitions of hate speech.

5.2 Predictive Entropy

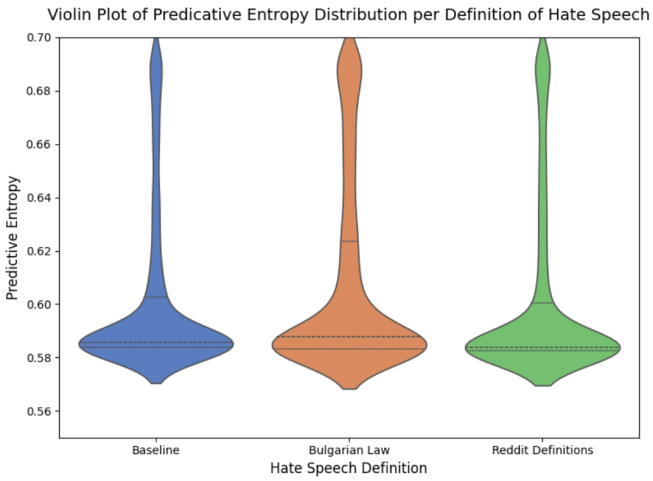

Figure 3: Predictive Entropy violin plot at temperature $T = 2.0$ with $k = 50$ samples

For Figure 3, a temperature of 2.0 was chosen because it is near the upper limit of the typical temperature scaling, according to [3]. This can be used to compare to the optimal calculated temperatures, to put the values into perspective. This violin plot shows that the variance of the predictive entropy is low, as most of the values are close to the mean. Additionally, the mean is similar among definitions, which shows that the total uncertainty at this temperature is similar.

Figure 4 shows that using the optimal temperatures makes the variance of the PE values increase significantly, especially for the Bulgarian definition. This is clear from the violin plot becoming broader, showing that the distribution is now more spread out. The Reddit definition was also affected, and the baseline was affected the least. The Bulgarian definition specifically has a bulge at the top, which shows that a large portion of the dataset got pushed towards near maximum uncertainty. This indicates that temperature scaling led to a large increase in test cases the model was highly uncertain about.

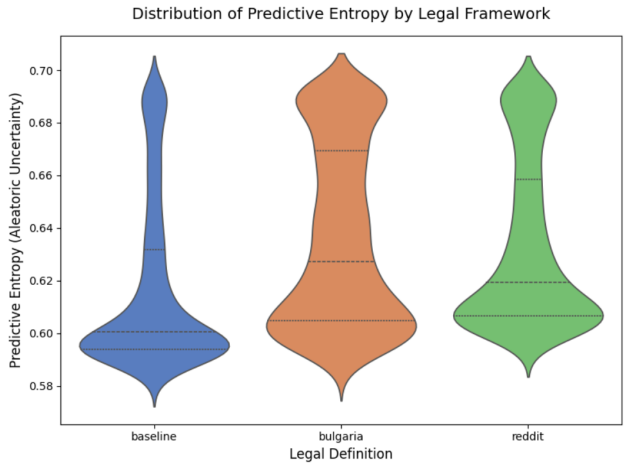

Figure 4: Predictive Entropy violin plot for optimal temperature with $k = 50$ samples.

This shows that the increase in temperature did not just create more outliers to increase the mean, but that a large number of test cases became highly uncertain as a result of temperature scaling. This shows one of the major disadvantages of temperature scaling with such high temperatures. While it did help improve the calibration according to ECE, it also made the uncertainty worse. More specifically, it made the model highly uncertain about test cases it was more certain about with lower temperature scaling.

Interesting to note is that this is not solely due to the increased temperatures, since the Bulgarian definition has a lower optimal temperature than the Reddit definition, but experienced a larger increase in the mean, alongside higher variance of PE.

A possible explanation for this is that the Bulgarian definition conflicted more with the LLM's pre-programmed idea of hate speech, which caused more uncertainty when the LLM tried to adapt.

5.3 Expected Calibration Error

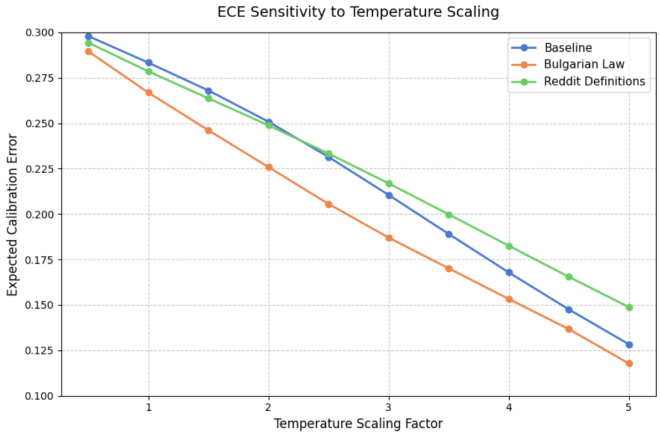

Figure 5: ECE plotted against temperature for $k = 50$ samples.

Figure 5 shows that the uncalibrated model at $T = 1$ has high ECE values across all definitions, compared to a common threshold of 5% or 0.05. This overconfidence before temperature scaling is a known phenomenon. As demonstrated by [4], models that are optimized using negative-log likelihood tend to overfit to probabilistic error. The result of this is that the logits are inflated to improve accuracy, at the cost of generalizing the model's calibration. Aggressive temperature scaling is required to bring the ECE score down as seen in the graph.

The graph also shows that the temperature scaling for optimal ECE does not match the optimal temperatures found by minimizing the negative log-likelihood (Table 1), as the global minima of the graphs are beyond $T = 5.0$. While ECE significantly flattens the distributions to optimize for the average, NLL takes a more balanced approach to optimization to keep the distribution meaningful. This shows that NLL is better at preserving a model's sharpness than ECE. Higher sharpness means more decisive output probabilities closer to 0% or 100%.

5.4 Brier Score

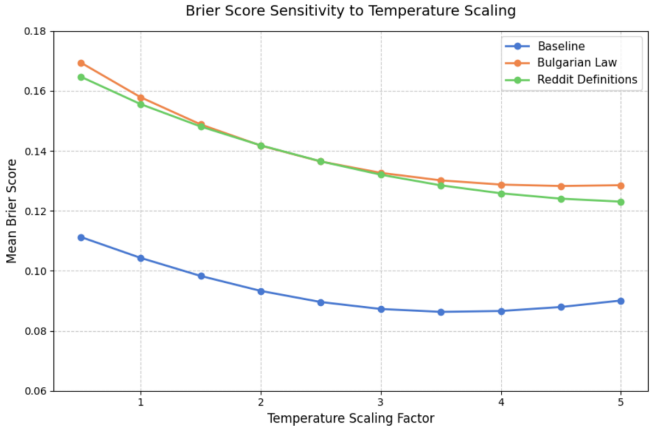

Figure 6: Mean Brier score plotted against temperature for $k = 50$ samples.

As demonstrated in Figure 6, the mean Brier scores for the explicit definitions are very similar. The baseline, on the other hand, is significantly lower. Despite the drop for the explicit definitions, all Brier scores are in a healthy range, which shows that the model is capable of the task of hate speech classification using these definitions.

Since Brier Score measures uncertainty and calibration, this shows that adding the definitions made the calibration and uncertainty worse. This supports the results from Figure 3 and Figure 4, which showed worse uncertainty when adding definitions of hate speech.

The graph also shows that temperature scaling does help improve the calibration. All definitions' Brier scores improved beyond the uncalibrated logits at $T = 1$. This shows that despite the downsides of aggressive temperature scaling, it helped further improve the model calibration.

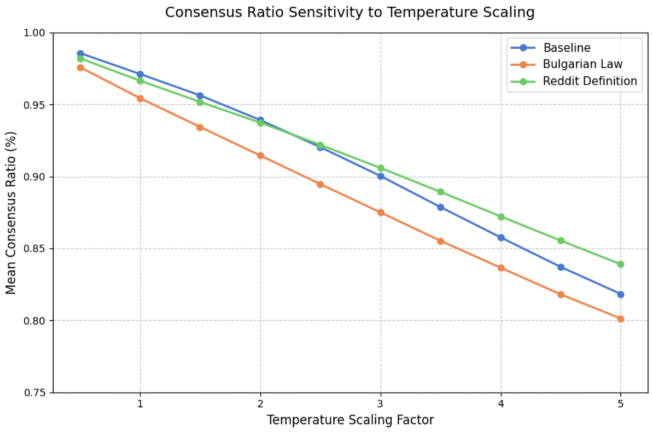

Figure 7: Mean consensus ratio plotted against temperature for $k = 50$ samples.

5.5 Consensus Ratio

Figure 7 is very similar to Figure 5, showing a trade-off between calibration and micro-consistency. As the temperature scaling factor increases, the ECE score (calibration) improves, but due to increased randomness, the CR score (micro-consistency) declines. Note that this trade-off is only present before the global minimum of the ECE score, which in this graph is beyond $T = 5.0$, after which both metrics will start declining.

Another reason that the graphs are so similar is related to the overconfident logits. Since both metrics are dependent on the logits, and the logits have high sharpness, less variation is possible in the data. This means that similar metrics measuring the related concepts of consistency and calibration will have less variation in the results as well.

Interestingly, while the Bulgarian definition has the best ECE scores, it has the worst mean CR scores. This is likely because the Bulgarian definition had the most uncertainty, meaning it was less confident about its predictions, resulting in lower logits. This explains the better ECE scores, because it is mathematically more difficult to calibrate a dataset that contains output probabilities with extreme sharpness.

Since the Bulgarian definition has the highest mean PE and uncertainty, it also has the flattest distribution of output probabilities. This means that a better ECE value on its own is not enough to conclude that the distribution has improved, as it may simply have been flattened. This shows that the best ECE score alone does not mean best calibration, as it may simply be the result of a flattened distribution.

Finally, in Figure 7 the gap between the Bulgarian definition and the other definitions starts small, increases as the temperature scaling factor goes up, then decreases again. This shows that the Bulgarian definition suffered the worst decline in micro-consistency as more aggressive temperature scaling was applied, when compared to the other definitions.

5.6 Overview and Statistical Significance

Table 2 shows that the baseline outperformed both explicit definitions, because all metrics worsened compared to the

Metric	Definition	Mean	SD	t-statistic	p-value
BS ↓	Baseline	0.104	0.280	-	-
	Bulgaria	0.158	0.330	-10.0	< 0.001
	Reddit	0.156	0.336	-11.4	< 0.001
PE ↓	Baseline	0.591	0.024	-	-
	Bulgaria	0.596	0.030	-8.6	< 0.001
	Reddit	0.592	0.026	-2.5	0.011
CR ↑	Baseline	0.971	0.085	-	-
	Bulgaria	0.954	0.107	8.1	< 0.001
	Reddit	0.967	0.095	2.6	0.010
mCON ↑	Baseline	0.958	0.114	-	-
	Bulgaria	0.934	0.139	8.7	< 0.001
	Reddit	0.952	0.123	2.4	0.017

Table 2: Paired t-tests comparing baseline to explicit definitions for all metrics at $T = 1.0$ ($k = 50$, $df = 49$).

baseline. The statistical analysis showed that the deviations from the baseline definition are statistically significant ($p < 0.05$). Despite this, metrics such as mCON reacted differently depending on the definition. While for the Reddit definition there is a minor decline in micro-consistency, the Bulgarian definition was impacted significantly more. This shows that the effect that the explicit definition has on the metrics may depend on which definition was used, and may also depend on the metric itself.

5.7 Spearman's Rank Correlation

Definition	Metric Pair	Correlation (r)
Baseline	BS vs. PE	0.96
	BS vs. mCON/CR	-0.58
	PE vs. mCON/CR	-0.67
Bulgarian	BS vs. PE	0.86
	BS vs. mCON/CR	-0.60
	PE vs. mCON/CR	-0.77
Reddit	BS vs. PE	0.89
	BS vs. mCON/CR	-0.52
	PE vs. mCON/CR	-0.68

Table 3: Spearman rank correlations (r) per definition.

The results show that CR and mCON have a correlation coefficient of 1.0 for every definition, which is why they are grouped together in Table 3. This shows that both metrics provide an identical ranking of model micro-consistency across all test cases for all definitions.

There is a strong negative correlation between mCON/CR and the uncertainty metrics PE and BS. This shows that higher micro-consistency corresponds with a lower uncertainty and better calibration (lower miscalibration), which supports the previous results. Finally, there is a very strong positive correlation between BS and PE, which shows that better calibration corresponds with lower uncertainty. This correlation is expected because both metrics capture uncertainty, and lower uncertainty allows the model to make more informed decisions about the token probabilities or logits.

As shown in Table 3, the magnitude of the correlations remains close between definitions, and the signs (+ or -) remain fully consistent. This shows that these correlations are mostly caused by the model's behavior, and that the definitions have an impact on the magnitude.

6 Discussion

6.1 ECE and Brier Score

As stated by [1] and [9], the ECE metric has multiple issues. These include high sensitivity to the number of bins leading to volatile results, bins canceling each other out due to only considering bin averages, and ignoring a large chunk of the probability space due to a biased spread of logits over the bins. Due to the sharpness of the logits, the spread of the logits is biased towards the final bin, meaning this will affect the ECE metric in our case.

ECE only looks at the maximum of the probability vector, while the Brier score looks at all the output probabilities. This means the Brier score provides a more informed evaluation, making it a better choice than ECE. Because of this, the results from Brier score should be given more weight than the ECE results.

6.2 Batching

In the original setup, the test cases were fed to the LLM one by one. The main limitation here was that this made the model so overconfident that 3513 out of 3728 test cases (94.2%) had $-\infty$ for one of the logits. A value of $-\infty$ is caused by the probability assigned to a token being near zero. This also caused some minor instability with classification despite using greedy decoding, which made the micro-consistency on deterministic reruns 98.2% instead of the expected 100%. This shows that greedy decoding does not equal determinism, as described by [14].

To fix this issue, the test cases were passed in batches. To ensure good execution speed while taking into account memory constraints, a batch size of 32 was chosen. This completely fixed the $-\infty$ logit issue. While the model was still overconfident, this reduced the overconfidence enough to provide more usable logits for the calculations of the metrics. These changes make the transition to batching an improvement for reproducibility. This also improves execution speed, which helped analyze more data due to time constraints.

6.3 Limitations

The experiments demonstrated that the results are volatile and dependent on the setup. Configuration details such as batching and float precision can have a large effect on the LLM predictions and logits. This means more research needs to be done to show that the results are a general property, and not just the behavior of the Llama-3-8B-Instruct model specifically.

One limitation of the current setup is the overconfident logits. This makes it hard to introduce variance in the data, which caused some graphs of related metrics to become very similar in the results. This does not mean temperature scaling failed to calibrate the logits, but rather it shows a flaw in the baseline confidence of the LLM. It is something that cannot

be calibrated post-hoc and requires better tuning before the prompting phase.

Another limitation is the use of stochastic label sampling. Currently, the reruns are simulated probabilistically instead of prompting the LLM multiple times, which is unable to isolate epistemic uncertainty. This is because epistemic uncertainty requires the model to process the data multiple times. Due to this loss of information, metrics such as mutual information cannot be calculated. Since the main focus is on the model's uncertainty, which is captured by epistemic uncertainty, this is an important limitation to consider.

It is also important to consider the possibility of data leakage, which means that the Llama-3-8B-Instruct model may have seen the HateCheck dataset before. This could partially explain the overconfidence of the model, along with the high consensus ratio at lower temperatures ($T \leq 2.0$). The main reason that this could be problematic is that it would mean that the experiments are measuring the performance of the memorization of the model instead of the zero-shot performance, which is less representative of real hate speech detection algorithms.

Finally, while system prompts are typically a good choice to provide LLM instructions on a task, in this case, it was insufficient to override the default programming. This is shown by the results in Figure 4, where the explicit definitions caused more uncertainty compared to the baseline.

6.4 Future Work

For future work, instead of using stochastic label sampling, Monte Carlo softmax sampling could be used. This allows calculating metrics like mutual information, which provide information about the epistemic uncertainty of the model.

Other aspects worth exploring are modifications in the setup. Some examples mentioned by [14] are using different precision modes like FP32, and trying different batch sizes. Other promising alternatives include trying different LLMs (such as encoder-decoder models) or experimenting with different prompts. Using LoRA could also drastically improve the quality of the LLM evaluations, since some definitions seem to interfere with the internal definition of hate speech. LoRA could overcome this by fine-tuning the model for the task of Hate Speech classification, instead of trying to override existing programming.

Trying more definitions would also be valuable to find patterns in the definitions that increase or reduce the stability of the LLM evaluations.

7 Conclusion

In conclusion, providing explicit definitions of hate speech for the task of hate speech classification using zero-shot prompting made the micro-consistency and uncertainty worse. This means that the baseline without an explicit definition outperformed the explicit definitions of hate speech. Some definitions, such as the Bulgarian definition, are more sensitive to temperature scaling when calibrating the model. This is likely due to a conflict between the explicit definition and the LLM's idea of hate speech. Another finding is that there is a trade-off between micro-consistency and calibration. The results also show that micro-consistency and uncertainty are strongly correlated.

However, more research needs to be done to show that the degradation of micro-consistency and uncertainty is caused by a fundamental challenge for LLMs when applying new definitions of hate speech for the task of hate speech classification. By extending the evaluations to other LLMs and configurations, future work can establish whether the findings represent a fundamental property of LLM-based classification, rather than just the evaluation of the Llama-3-8B-Instruct model.

The main takeaway for researchers in this field is that LLM configuration is incredibly important for the results, as even minor modifications can have major impact on the outcome. Having sufficient data is vital to prove that the results show a general property and not just a model evaluation. More research needs to be done with alternative LLMs, configurations or the use of LoRA to definitively confirm the negative effect of explicit hate speech definitions on hate speech classification. However, current findings still support the idea that explicit definitions of hate speech caused worse micro-consistency and uncertainty.

Responsible Research

Code of Conduct

This research was conducted following the Netherlands Code of Conduct for Research Integrity described in [13]. The principles of honesty, scrupulousness, transparency, independence and responsibility were followed.

Data and Privacy

Many test cases used in the HateCheck dataset contain offensive language. This dataset was used for educational purposes only. Using such a dataset also allowed the research to avoid using Personally Identifiable Information (PII), such as specific hateful tweets that can be tracked back to the user.

Biases

The Llama-3-8B-Instruct model and the HateCheck dataset labels come with biases. For stability analysis, this can act as a confounding variable when the LLM classifications are compared to the labels and used to determine correctness. This can also have an impact on how representative the dataset is of hate speech, as it may exclude lesser-known forms of hate speech. While the primary objective of this research is to analyze the stability, there is an inductive risk in the methodology related to representation. Thus, our findings should be interpreted with these limitations in mind.

Reproducibility and Replicability

To ensure replicability, all formulas, libraries, prompts, parameters, and other configuration details have been disclosed. The exact steps taken are carefully described in the methodology. Achieving perfect replicability may not be possible due to the volatility of BF16, as shown by [14]. However, due to sampling, the average results should still be sufficiently close for reproducibility.

Use of AI

Generative AI was used to assist with specific tasks, not to automatically generate the content. This was used to assist in the process of brainstorming ideas, assist with writing code, and to enhance the quality of writing. All AI-generated content was manually checked and verified before being used.

Impact

We recognize the dual-use potential of our findings. While the objective was to analyze the stability to improve future use of LLMs for hate speech classification, the findings could be exploited to bypass hate speech detection. The positive use case is improving the use of LLMs for hate speech detection, allowing fair and reliable use. Alternatively, knowledge about limitations of LLMs for hate speech detection can lead to improving models or configurations further before deploying hate speech detection algorithms in real-life scenarios. This helps prevent reinforcing biases in existing models, and helps establish a roadmap of improvements to work on in the future.

References

[1] Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernandez. Stop measuring calibration when humans disagree. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1892–1915, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[2] Nghia T Bui, Guergana Savova, and Lijing Wang. Assessing the macro and micro effects of random seeds on fine-tuning large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2024.

[3] Comet ML. Llm parameter optimization: Understanding temperature and top-p. https://www.comet.com/site/blog/llm-parameter-optimization/, 2023. Accessed: 2024-05-15.

[4] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pages 1321–1330. PMLR, 2017.

[5] Eyke Hüllermeier and Willem Waegeman. Aleatory and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

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

[7] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic entropy probes for epistemic uncertainty in large language models. In *International Conference on Machine Learning (ICML)*, pages 17822–17840. PMLR, 2023.

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

[9] Jeremy Nixon, Michael W Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. Measuring calibration in deep learning. *arXiv preprint arXiv:1904.01685*, 2019.

[10] Paul Röttger, Bertram Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. HateCheck: Functional tests for hate speech detection models. 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 41–58, Online, August 2021. Association for Computational Linguistics.

[11] Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[12] Dennis Ulmer, Tuan Hoang Tran, Christian Röttger, and Detmar Meurers. Know what you don't know: Unsupervised accuracy estimation for classification. *arXiv preprint arXiv:2210.15452*, 2022.

[13] Universiteiten van Nederland (UNL), Netherlands Organisation for Scientific Research (NWO), and Royal Netherlands Academy of Arts and Sciences (KNAW). Netherlands code of conduct for research integrity, 2018.

[14] Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Understanding and mitigating numerical sources of nondeterminism in llm inference. In *Advances in Neural Information Processing Systems*, 2025.

A Experimental Setup

A.1 Environment and libraries

This experiment ran on Kaggle using their provided 2x Nvidia T4 GPUs, using the Hugging Face transformers library. For calculations NumPy, pandas, SciPy (for minimize), scikit-learn (for log_loss) and nlp-uncertainty-zoo (for PE and ECE) were used. Other relevant tools are bitsandbytes (0.49.2), transformers (5.10.2) and accelerate (1.13.0).

A.2 Configuration and batching

Listing 1: pipeline configuration

```
from transformers import BitsAndBytesConfig

quantization_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_compute_dtype=torch.bfloat16,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_use_double_quant=True
)

model_id = "meta-llama/Meta-Llama-3-8B-Instruct"
hf_token = user_secrets.get_secret("HF_TOKEN")

model = AutoModelForCausalLM.from_pretrained(
    model_id,
    device_map="auto",
    quantization_config=quantization_config,
    token=hf_token
)

pipe = pipeline(
    "text-generation",
    model=model,
    tokenizer=tokenizer,
    device_map="auto",
    clean_up_tokenization_spaces=False,
)
```

Listing 2: Batch prompting and output generation

```
prompts = [
    tokenizer.apply_chat_template(
        [{"role": "system", "content": system_content}, {"role": "
     user", "content": msg}],
        tokenize=False,
        add_generation_prompt=True
    ) for msg in batch_messages
]

inputs = tokenizer(prompts, return_tensors="pt", padding=True).to(
     "cuda")

with torch.inference_mode():
    outputs = model.generate(
        **inputs,
        max_new_tokens=1,
        output_scores=True,
        return_dict_in_generate=True,
        do_sample=False  # deterministic
    )
```

B Definitions

B.1 Reddit Definition

Rule 1: Remember the human. Reddit is a place for creating community and belonging, not for attacking marginalized or vulnerable groups of people. Everyone has a right to use Reddit free of harassment, bullying, and threats of violence. Communities and people that incite violence or that promote hate based on identity or vulnerability will be banned. Marginalized or vulnerable groups include, but are not limited to, groups based on their actual and perceived race, color, religion, national origin, ethnicity, immigration status, gender, gender identity, sexual orientation, pregnancy, or disability. These include victims of a major violent event and their families. While the rule on hate protects such groups, it does not protect those who promote attacks of hate or who try to hide their hate in bad faith claims of discrimination. Some examples of hateful activities that would violate the rule: - Community dedicated to mocking people with physical disabilities. - Post describing a racial minority as sub-human and inferior to the racial majority. - Comment arguing that rape of women should be acceptable and not a crime. - Meme declaring that it is sickening that people of color have the right to vote. - Post promoting harmful tropes or generalizations based on religion (e.g. a certain religious group controls the media, or consists entirely of terrorists). - A comment denying or minimizing the scale of a hate-based violent event. Additionally, when evaluating the activity of a community or an individual user, we consider both the context as well as the pattern of behavior.

B.2 Bulgarian Definition

Article 162 (1): Anyone who, by speech, press or other media, by electronic information systems or in another manner, propagates or incites discrimination, violence or hatred on the grounds of race, nationality or ethnic origin shall be punishable by imprisonment of one to four years and a fine from BGN 5,000 to 10,000, as well as public censure. Article 164 (1): A person who propagates or instigates discrimination, violence or hatred on religious basis by speech, through the press or other mass media, through electronic information systems or in another way, shall be punished by imprisonment for up to four years or probation and a fine from BGN five thousand to ten thousand.

C Prompts

C.1 Binary Classification Prompt

Respond ONLY with '1' if you consider it Hate Speech or '0' if you don't. No prose, only a '0' or a '1'

C.2 Modified Binary Classification Prompt

Respond ONLY with '1' if you consider it Hate Speech or '0' if you don't. If you refuse to answer, respond with 'X'. No prose, only a '0', '1' or X