Proceedings of Machine Learning Research 230:1–18, 2024 Conformal and Probabilistic Prediction with Applications

Calibrated Large Language Models for Binary Question Answering

Patrizio Giovannotti PATRIZIO.GIOVANNOTTI.2019@LIVE.RHUL.AC.UK
Royal Holloway, University of London, Egham, Surrey, UK
Centrica, UK
Alex Gammerman A.GAMMERMAN@RHUL.AC.UK
Royal Holloway, University of London, Egham, Surrey, UK

Editor: Simone Vantini, Matteo Fontana, Aldo Solari, Henrik Boström and Lars Carlsson

Abstract

Quantifying the uncertainty of predictions made by large language models (LLMs) in binary text classification tasks remains a challenge. Calibration, in the context of LLMs, refers to the alignment between the model's predicted probabilities and the actual correctness of its predictions. A well-calibrated model should produce probabilities that accurately reflect the likelihood of its predictions being correct. We propose a novel approach that utilizes the inductive Venn–Abers predictor (IVAP) to calibrate the probabilities associated with the output tokens corresponding to the binary labels. Our experiments on the BoolQ dataset using the Llama 2 model demonstrate that IVAP consistently outperforms the commonly used temperature scaling method for various label token choices, achieving well-calibrated probabilities while maintaining high predictive quality. Our findings contribute to the understanding of calibration techniques for LLMs and provide a practical solution for obtaining reliable uncertainty estimates in binary question answering tasks, enhancing the interpretability and trustworthiness of LLM predictions.

Keywords: large language models, calibration, uncertainty estimation, binary question answering, Venn–Abers predictor

1. Introduction

Language models have evolved dramatically, progressing from simple n-gram models to large pre-trained neural networks based on the transformer architecture (Vaswani et al., 2017). However, their core task remains predicting the next word in a sequence given the previous context. This rudimentary capability has proven remarkably versatile when combined with prompting techniques that allow language models to perform diverse tasks simply by modifying the input text.

For instance, to predict a film review's sentiment using a large language model (LLM), one could construct a prompt:

> *Read the following review: [...] The reviewer's opinion is mostly*

By continuing this prompt, an LLM can generate words like "positive" or "negative", effectively performing binary sentiment classification without being explicitly trained on that task. This *zero-shot* capability of modern LLMs is powerful, but comes with a critical challenge – how to reliably quantify the uncertainty of their predictions?

While state-of-the-art LLMs excel at generating fluent and relevant text, their underlying sequence-to-sequence nature makes uncertainty estimation non-trivial. This work proposes a simple yet effective approach to extract well-calibrated uncertainty estimates from LLMs for binary question answering tasks, without any further model training or modifications.

The key idea is to directly calibrate the raw word scores (logits) produced by the LLM during text generation. We focus on the logits corresponding to the binary class labels (e.g. "yes" and "no") at the first step of generation. By applying Venn–Abers predictors (Vovk et al., 2022; Vovk and Petej, 2014) – a type of conformal predictor providing calibration guarantees under the i.i.d assumption – we learn an optimal isotonic mapping between these logits and calibrated class probabilities.

We demonstrate the effectiveness of our approach on two binary question answering datasets using the open-source LLM Llama 2 7B (Touvron et al., 2023). A key advantage is that no further model training – i.e. any modification to the model's weights as a result of observing examples relevant to the task – is required, making our method a zero-shot solution for uncertainty-aware binary text classification with LLMs. We also compare against temperature scaling (Guo et al., 2017) and show improved calibration performance.

The remainder of this paper is structured as follows: Section 2 provides background information, Section 3 describes the proposed methodology in detail, Sections 4 and 5 present the experimental setup and results, Section 6 comments related work, Section 7 concludes the paper and outlines potential future research directions.

2. Background

Formally, the language modelling task (see Jurafsky and Martin, 2009) is to compute the probability of a given sequence of words $P(w_{1:n}) = P(w_1, w_2, \ldots, w_n)$, $w_i \in W \; \forall i = 1, \ldots, n$. This relies on computing the probability of each word w_i given the previous words:

$$P(w_{1:n}) = \prod_{i=1}^{n} P(w_i \mid w_{1:i-1}) \; .$$

Estimating directly such a probability is impossible, given the diversity and continuous evolution of human language; however, there are many ways to approximate its value: the simple *bigram* model, for instance, is based on the Markov assumption $P(w_i \mid w_{1:i-1}) \approx P(w_i \mid w_{i-1})$, with the right-hand side calculated as the proportion of occurrences of the word w_i following word w_{i-1} in a large corpus of text. Current state-of-the-art models for language modelling and text generation, on the other hand, use large *decoder* architectures which are pre-trained on predicting the next word over massive text corpora. Built upon the attention mechanism (Sutskever et al., 2014) and often requiring the learning of billions of parameters, decoders were introduced as a component of the first transformer architecture (Vaswani et al., 2017), but quickly grew to become the foundation of many successful autoregressive generative models, such as the GPT family (Radford et al., 2019). By effectively estimating the conditional probabilities of words following a given context, generative models are flexible enough to generate coherent text in response to a prompt submitted by the user.

Tokenization For text modelling purposes, language have too many words: sampling one out of all possible words of a language at every step is computationally demanding and does not address the presence of unknown words. Instead, LLMs consider sub-words, also known as **tokens**, that can be part of a word or full words, depending on their frequency in a large reference corpus of text. For example, using SentencePiece, Llama 2's default tokenizer (Kudo and Richardson, 2018), the word "positive" is represented as a single token `_positive`, where the underscore indicates a whitespace preceding the tokenized word. The word "Positive", on the other hand, is not frequent enough to deserve its own token, so it is represented as two consecutive tokens `_Pos` + `itive`. This strategy helps keep the vocabulary size K manageable, as less frequent words can be represented using combinations of known tokens rather than requiring dedicated ones, but it introduces a layer of complexity whenever we want to use the tokens for other purposes, such as text classification.

Language models as text classifiers Let $\mathcal{W} = \{w^{(1)}, \dots, w^{(K)}\}$ be a vocabulary of tokens: for example, we could consider the set of all English words post-tokenization. At each text generation step, an LLM outputs a vector $\boldsymbol{u} \in \mathsf{R}^K$, where each component u_k – called a *logit* – represents the unnormalized log-probability of token $w^{(k)}$ being the next token in the sequence. These logits can be converted into a probability distribution over the full vocabulary using the softmax function:

$$P(w^{(k)} \mid \boldsymbol{x}) = \frac{\exp(u_k)}{\sum_{j=1}^{K} \exp(u_j)} \tag{1}$$

where $\boldsymbol{x}$ is the input sequence. The next token to be generated is then chosen based on a decoding strategy, such as greedy search, beam search, or sampling methods like top-k or nucleus sampling (Holtzman et al., 2020).

To use an LLM as a classifier, we can provide a list of tokens representing potential labels and prompt the model to select the token that best classifies the input text. However, due to the stochastic nature of text generation, there is no guarantee that the LLM will actually output one of the specified labels, especially for smaller models like Llama 7B.

To address this issue, we propose directly extracting the logits u_k corresponding to the LLM's output tokens at the first step. For example, in a binary question answering task, we extract the logits for tokens representing "Yes" and "No", or alternatively their softmax values. We will refer to these tokens as *answer-tokens*. Although these scores are an indication of the token's likelihood to be the true label (or answer), they cannot be directly interpreted as well-calibrated probabilities, since softmax does not guarantee any validity or calibration property.

Calibration We refer to Guo et al. (2017) and define calibration in the following way: given a prediction $\hat{Y}$ for the label Y, returned with an estimated confidence $\hat{P}$, an ML model is *perfectly calibrated* if

$$\mathsf{P}(\hat{Y} = Y \mid \hat{P} = p) = p, \qquad \forall p \in [0, 1]$$

For instance, let us assume our model made 100 predictions, each with estimated probability $\hat{P} = 0.75$. If the model is perfectly calibrated, exactly 75 out of those 100 predictions need to be correct. In our scenario, a well-calibrated model would output probabilities for the "Yes"

token that reflect the true rate of positive labels in the test set. Simply applying a softmax function to the raw logits is not enough in most cases, and predictions from LLMs are often poorly calibrated. Moreover, using softmax does not provide a measure of confidence in the probability estimates themselves – a property generally enjoyed by imprecise probabilities (Destercke, 2022).

To overcome these limitations, we employ a recently developed calibration method, which we describe in the following section.

3. Methodology

We discuss the two methods we considered to obtain calibrated probabilities as a form of uncertainty estimation: Venn–Abers prediction and temperature scaling. There are many other calibration methods, such as Platt scaling or traditional isotonic regression, however temperature scaling is a popular and widely used technique in the deep learning setting, so we believe it may be the most appropriate competitor for our approach.

3.1. Venn–Abers Predictors

The core of our methodology revolves around the use of Venn–Abers predictors for calibration. Venn–Abers predictors (Vovk et al., 2022), a statistical tool used for probabilistic predictions, are employed to adjust the confidence levels of the LLMs' outputs. We detail the mathematical foundation of these predictors and how they are applied to calibrate the models.

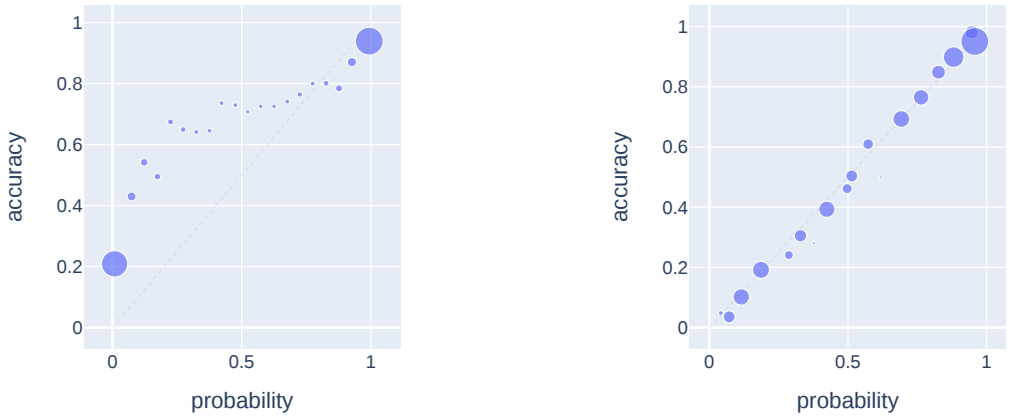
Venn–Abers predictors (Vovk and Petej, 2014) are a special case of Venn predictors, a class of probabilistic predictors guaranteed to be valid under the sole assumption of the training examples being exchangeable. Like all Venn predictors, they hold their validity guarantee and output multiple probability distributions over the labels – one for each possible label. The validity property implies perfect calibration (see Figure 1 for a graphic depiction of a valid model vs a not valid one). It has been proven that it is impossible to build a valid probabilistic predictor, in the general sense (Gammerman et al., 1998).

As an alternative to the definition given in Section 2, calibration can be interpreted as follows: let the random variable $Y \in \{0, 1\}$ model the label predicted by a binary classifier. Let $P \in [0, 1]$ be the confidence associated to the same prediction. P is perfectly calibrated if for the conditional expectation E

$$\mathsf{E}(Y|P) = P$$

almost surely.

Venn–Abers predictors (VAPs) are binary predictors and output a pair of probabilities (p_0, p_1) for each test example (x, y). The former is the probability of $y = 1$ should the true label be 0, while the latter is the probability of $y = 1$ should the true label be 1: one of the two is the valid prediction, but we don't know which one (as we don't know y). Because we always have $p_0 < p_1$, the pair (p_0, p_1) can be interpreted as the lower and upper probabilities, respectively, of a certain prediction. Depending on the test example, p_0 and p_1 may be more or less different in magnitude, although they are usually close to each other. A large gap between p_0 and p_1 signifies low confidence in the probability estimation –



(a) (b)

Figure 1: Reliability charts for (a) Llama 2 7B evaluated zero-shot on our BoolQ test set and (b) inductive Venn–Abers predictor based on the same model. The size of the circles represent the proportion of dataset observations falling in a given bin.

something traditional probabilistic predictors are not able to provide. For practical reasons however, it is often useful to have one probability estimate per test example. A reasonable way to combine the two numbers, as explained in Vovk and Petej (2014), is to calculate the probability which minimizes the regret for the log loss function:

$$p = \frac{p_1}{1 - p_0 + p_1} \,.$$

In this work we will be using the *inductive* variant of VAPs (IVAP), which was proposed as a computationally lighter version of VAPs in Vovk et al. (2015). This is our only option as the traditional VAP needs to be retrained for each test example, something absolutely infeasible given the average training time of a transformer model. The only difference with the classical IVAP is that we do not require a proper training set, since the underlying algorithm is pre-trained. This also means we can make use of much more data for calibration and testing.

An IVAP can be created as follows. Suppose we have a binary classification problem and a *scoring algorithm*, i.e. any ML algorithm that can issue a confidence score for each prediction – in our case, a pretrained transformer M. The dataset can be seen as a sequence of n objects x_i labelled as y_i, that is, $\mathcal{D} = (x_1, y_1, \ldots, x_n, y_n)$. We divide $\mathcal{D}$ in a calibration set $\mathcal{C}$ of size m and a test set $\mathcal{T}$ of size $n - m$. We run M over all examples in $\mathcal{C}$ and obtain m raw scores (for example, logits of the answer-tokens). For each test object x_j in $\mathcal{T}$, we predict a score z_j using M and append it to $\mathcal{C}$; then, we fit one isotonic regression on the augmented $\mathcal{C}$ for the case $y_j = 0$ and one for $y_j = 1$. The resulting probabilities $(p_0, p_1)_j$ are returned for observation x_j.

The general procedure to fit an IVAP is given in Algorithm 1 (see also Johansson et al., 2021). Isotonic regression is a nonparametric form of regression that fits a step-wise, non-decreasing function to a set of examples (see Zadrozny and Elkan, 2002). IVAPs still require for the isotonic regression to be re-calculated for each test example, for each label. However, Vovk et al. (2015) designed an optimised version that requires a single pre-calculation step, then performs an efficient evaluation step for every test example. We use an implementation written in Python.[1]

Algorithm 1: Pretrained inductive Venn–Abers predictor

Input: Dataset $\mathcal{D} = (x_1, y_1, \ldots, x_n, y_n)$; pretrained model M; calibration size m
Output: Multiprobabilities $((p_0, p_1)_{m+1}, \ldots, (p_0, p_1)_n)$
create calibration set $\mathcal{C} = (x_1, y_1, \ldots, x_m, y_m)$ from $\mathcal{D}$
create test set $\mathcal{T} = (x_{m+1}, y_{m+1}, \ldots, x_n, y_n)$ from $\mathcal{D}$
for $i \leftarrow 1$ **to** m **do**
  compute score for positive label $z_i = M(x_i)$
end
for $j \leftarrow m + 1$ **to** n **do**
  compute score for positive label $z_j = M(x_j)$
  fit one isotonic regression f_0 on the set $(z_1, y_1), \ldots, (z_m, y_m), (z_j, 0)$
  fit one isotonic regression f_1 on the set $(z_1, y_1), \ldots, (z_m, y_m), (z_j, 1)$
  produce the multiprobability $(p_0, p_1)_j = (f_0(z_j), f_1(z_j))$
end

3.2. Temperature scaling

The softmax function described in Equation 1 can be modified with an optional parameter τ, called the *temperature*, which is set in advance and can alter the softmax distribution. Let $\boldsymbol{u} = (u_1, \ldots, u_K)$ be the vector of logits returned by the LLM when predicting the next word w_i. The probability of word $w^{(k)}$ being chosen at step i is given by the temperature-scaled softmax:

$$P(w_i = w^{(k)} \mid w_1, \ldots, w_{i-1}) = \text{softmax}_\tau(u_k) = \frac{\exp(u_i/\tau)}{\sum_{k=1}^{K} \exp(u_k/\tau)} .$$

Smaller values of τ (i.e., $\tau < 1$) produce a sharper probability distribution, concentrating most of the probability mass on the most likely words. Conversely, larger values of τ (i.e., $\tau > 1$) result in a smoother distribution, assigning more probability to less likely words. When $\tau = 1$, the temperature-scaled softmax reduces to the standard softmax function.

Temperature scaling (Guo et al., 2017) is a popular calibration method in deep learning. It involves learning a temperature value $\hat{\tau}$ by minimising a calibration loss (e.g., negative log-likelihood) on a separate validation set. The learned parameter $\hat{\tau}$ is expected to approximate the optimal temperature τ^*, which minimises the calibration error on the test set.

1. https://github.com/ptocca/VennABERS

Temperature scaling is well-suited for deep learning because it employs the same training methodology as the main model and extends naturally to the multiclass setting.

However, temperature scaling has some limitations. Its effectiveness depends on how well the learned temperature $\hat{\tau}$ approximates the optimal temperature τ^*. This approximation relies on two key factors: the similarity between the validation and test distributions, and the effectiveness of the learning algorithm used to estimate $\hat{\tau}$. If the validation set is not representative of the test set, or if the learning algorithm fails to find a good approximation, the calibration performance may degrade. Furthermore, since temperature scaling is a linear transformation of the model's logits, it has an inherent limit on the level of calibration improvement it can achieve, especially if the model's initial calibration is poor.

In contrast, the Venn-Abers predictor always achieves the optimal calibration performance, *irrespective of the temperature*. This property is particularly valuable for LLMs, where users often adjust the temperature to control the generated text's creativity.

4. Experimental Setup

All the experiments are performed using the Llama 2 7B language model, released by Meta as the smallest of the Llama 2 family (Touvron et al., 2023). Llama 2 7B has a relatively small footprint: it needs about 14 GB of dedicated GPU RAM when making predictions in half precision (16 bit). Because our approach is zero-shot, there is no need for additional 14 GB of memory to store the model gradients for the training step. Most importantly, Llama 2 is an open-source model, and grants access to all its internal components and outputs – an essential feature of any *white-box* approach (see Section 6). The version used in this work is `meta-llama/Llama-2-7b-chat-hf`, available on Hugging Face, loaded on a single Nvidia A10G card.

4.1. Dataset

Boolean Questions (BoolQ – Clark et al., 2019) is a question answering dataset for yes/no questions which are produced spontaneously (without specific prompts or directions) by annotators reading a Wikipedia passage. Each example is a triplet $\langle \text{question}, \text{passage}, \text{answer} \rangle$, where the task is to answer a binary question related to the text passage.

In our zero-shot configuration, the original training set is shuffled with the original validation set (the test set is not publicly available), for a total of 12,697 examples. We retain 20% of it to separately train our Venn–Abers predictor and use the remaining 10,156 examples as test set.

Each example was edited into a prompt that could elicit a satisfactory response from the LLM. Given the relatively small scale of Llama 2 7B, the prompt has been kept as simple as possible. An example of prompt is the following:

> Context:
> "The Air Force usually does not have fighter aircraft escort the presidential aircraft over the United States but it has occurred, for example during the attack on the World Trade Center."
> Question: "Does air force one travel with fighter escort?"

Yes or No?
Answer:

4.2. Evaluation metrics

To evaluate calibration performance we use the Expected Calibration Error (ECE). To compute ECE (Naeini et al., 2015), all predictions are grouped in M bins of equal width, such that bin B_m contains examples with confidence ranging in $(\frac{m-1}{M}, \frac{m}{M}]$. ECE is defined as

$$\text{ECE} := \frac{1}{n} \sum_{m=1}^{M} |B_m| \cdot |p(B_m) - \hat{p}(B_m)|$$

where $p(B_m)$ is the true fraction of positive instances in bin B_m and $\hat{p}(B_m)$ is the average estimated probability for predictions in bin B_m. For example, an ECE of 0.10 means that on average, the models' expected probability for a prediction is off by 10%. It is important to note that ECE varies depending on the number of bins M: throughout our experiments we will report results for $M = 10$, which is standard practice in calibration studies – see for example Guo et al. (2017).

To specifically assess prediction quality, we use the area under the ROC curve (AUC), the curve obtained by plotting false positive rate against true positive rate at different classification thresholds. By using AUC, we measure the model's ability of ranking positive examples higher than negative examples, irrespective of the classification threshold and, consequently, irrespective of the model's calibration. Choosing fixed-threshold metrics such as F_1 or Matthews Correlation Coefficient would penalise uncalibrated models and hide its actual predictive power.

Additional evaluation metrics are defined and their associated results reported in Appendix A.

5. Results

Following the approach detailed in Section 2, we extract the logits for both our answer-tokens to predict a binary answer and, consequentially, train our Venn–Abers predictor. We consider two alternative transformations of these scores:

1. "Yes" and "No" scores selected from the softmax over all K logits (softmax-K)
2. Scores from softmax computed over the sole "Yes" and "No" logits (softmax-2)
3. Calibrated version of 1, via the inductive Venn–Abers predictor (IVAP-K)
4. Calibrated version of 2, via the inductive Venn–Abers predictor (IVAP-2)

through a range of temperature values. We consider two pairs of answer-tokens: (_Yes, _No) and (Yes, No). The underscore prefix in the first pair indicates that the token is considered a start-of-word token, while the tokens in the second pair can appear in any part of a word (see Section 2). This subtle distinction is specific to the tokenizer used: a different tokenizer may ignore white spaces and generate the same Yes token regardless of the word's context; in some cases, a token may not even be included in the vocabulary and no logit would

be produced as a result. Choosing the right answer-tokens is a delicate early step of our approach and may significantly impact a model's behaviour and performance.

We evaluate our calibration method using expected calibration error and AUC (see Section 4.2). In Appendix B we report results for a further NLP task, sentiment classification.

5.1. Calibration results

In terms of calibration performance, the advantage of using Venn–Abers predictors is evident. Figure 2 shows ECE values for both answer-token choices. When using start-of-word tokens (_Yes, _No), Softmax-K shows a minimum at a specific temperature ($\tau \approx 1.8$) but degrades rapidly as soon as we move away from it. Softmax-2, on the other hand, shows several local minima and a global one for $\tau \approx 33$ which outperforms the former model. For the alternative choice (Yes, No), Softmax-2 shows a global minimum at a relatively low temperature, while Softmax-K fails to calibrate the predictions and exhibits high ECE at any temperature.

In contrast, the Venn–Abers predictors achieve an excellent calibration performance for both token pairs, at any temperature, with the exception of very low values ($\tau < 1$) where all models seem to struggle (intuitively, lower temperatures push probabilistic predictions towards the extremes 0 and 1, hence there is little room for the scores to be adjusted).

These findings suggest that while temperature scaling can improve calibration in some cases, it is highly sensitive to the choice of temperature value and may not be effective for all token pairs. On the other hand, the Venn–Abers predictor offers a more reliable and consistent method for obtaining well-calibrated probabilities, making it a promising approach for uncertainty estimation in language models.

5.2. Prediction quality

We report in Figure 3 the AUC scores for all models and configurations. We observe immediately that both the original Llama 2 model and the calibrated model obtained via Venn–Abers prediction exhibit similar AUC scores across different temperature settings. This suggests that applying the Venn–Abers predictor does not significantly impact the model's ranking performance, preserving its ability to discriminate between positive and negative examples.

Again, Softmax-2 (and IVAP-2) outperform the two competitors and achieve high AUC for both answer-token choices; Softmax-K works better with the (Yes, No) pair, which unfortunately is the configuration where it scored the worse ECE. In contrast, IVAP-2 was well-calibrated.

Additionally, we note again that higher temperature values generally result in improved predictive performance, indicating that a more smoothed probability distribution is beneficial for this task.

6. Related Work

This work follows the original application of Venn–Abers predictors to pretrained transformers introduced by Giovannotti (2022), which we extend to the generative case. Our approach requires access to the internal components of the LLM, namely its output logits,

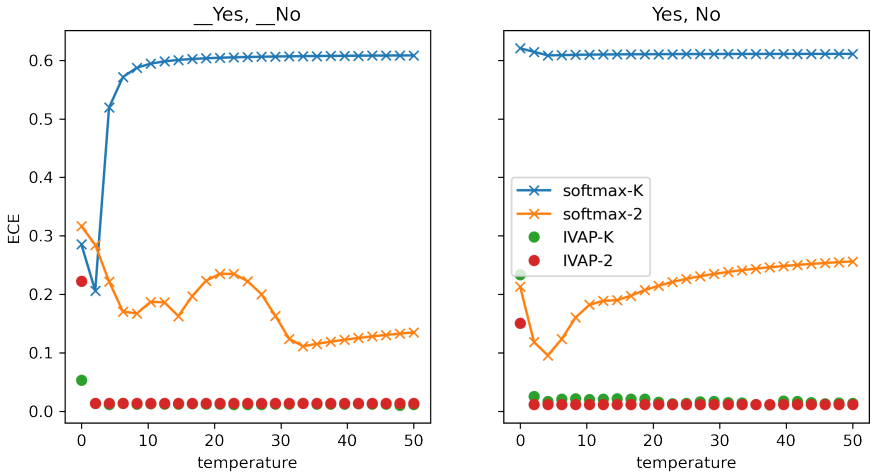


Figure 2: Expected calibration error of the original Llama 2 model and its Venn–Abers version (IVAP). Our IVAP results in consistently low errors and outperforms temperature scaling, whether we use as labels start-of-string tokens (left) or generic ones (right). IVAP is also invariant w.r.t. how many tokens are considered in the softmax (2 or K).

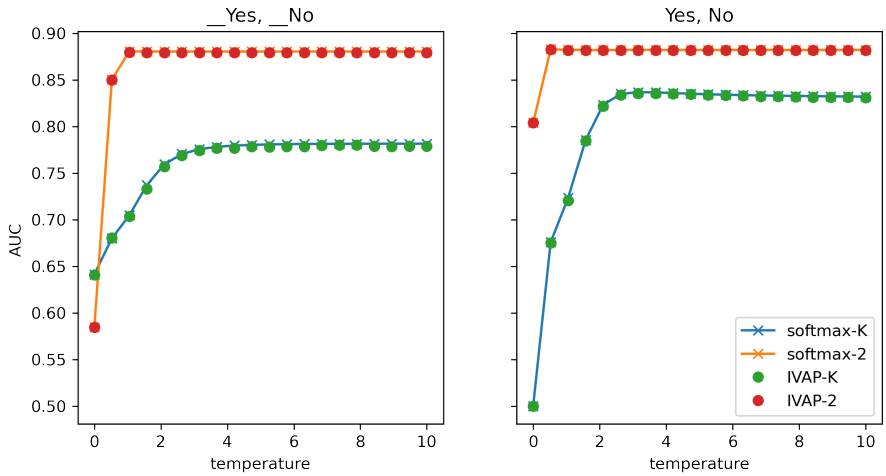


Figure 3: Area under the ROC curve computed at different temperatures for both models. A positive label was predicted by considering either start-of-word `_Yes` tokens (left plot) or generic `Yes` tokens (right plot).

and can be seen as a white-box approach to uncertainty quantification (UQ). GPTScore (Fu et al., 2023) is another example of white-box UQ that uses output token weights; other approaches consider the model's internal states (Azaria and Mitchell, 2023) or require a fine-tuning step to learn to express their uncertainty (Lin et al., 2022).

Conversely, black-box approaches do not require any knowledge of the model. Kapoor et al. (2024) propose a fine-tuning procedure that calibrates the model based on its own evaluation of the generated answer. Manakul et al. (2023)'s SelfCheckGPT computes a confidence score by comparing each LLM claim to N stochastically-generated responses. Together with Kadavath et al. (2022)'s, this work inspired Agrawal et al. (2024) to probe LLMs with different question templates for hallucination detection in the context of reference quotation. Kuhn et al. (2023) use an auxiliary model to cluster alternative responses by similarity, Ulmer et al. (2024a) employs an external model to compute a numerical confidence score, while CRITIC (Gou et al., 2024) can leverage a variety of external tools to validate its output.

Conformal prediction has been recently used in the context of LLM generation: Ravfogel et al. (2023) showed how to build output token sets containing the correct token at a rate $1 - \alpha$; Ulmer et al. (2024b) extended this *conformal nucleus sampling* strategy to the non-exchangeable case. Su et al. (2024) studied the application of CP to black-box models, that is, whenever no access to the logits is available. In machine translation, conformal prediction has been used to evaluate translation quality by Giovannotti (2023) and Zerva and Martins (2023).

7. Conclusion

We presented a competitive method to calibrate the output of large language models in the binary question answering setting. Our approach, based on inductive Venn–Abers predictors (IVAP), requires no further training of the LLM and does not require any special assumption on the distribution of the data.

Our experiments demonstrated that IVAP outperforms a temperature scaling approach and guarantees low calibration error over a broad temperature range. This also applies when choosing different tokens to represent the binary labels. In other words, our approach is invariant with respect to the temperature and to the answer-tokens of choice.

The natural continuation of our work would address question answering with more than two labels, or ideally *open* question answering, where answers can be made of any number of tokens. Additionally, it would be interesting to find the minimum calibration set size that would guarantee an acceptable performance: 1/4 of the test set size may still be too much in certain scenarios.

In conclusion, this is a first step towards a reliable and safer AI, where models can precisely determine and communicate their degree of uncertainty in relation to any answer.

Acknowledgements

This work was partially funded by Centrica plc. Thanks to Chris Watkins for clarifying some technical aspects, and to Ilia Nouretdinov for his suggestions and insight.

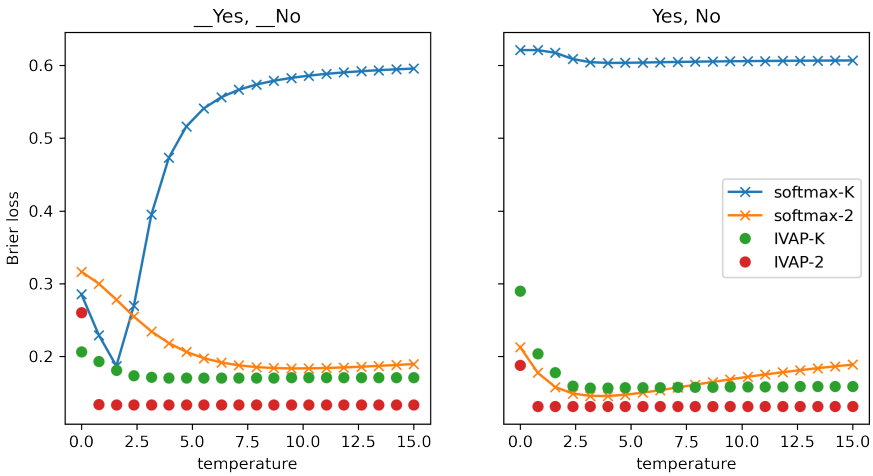


Figure 4: Brier loss for the original Llama 2 model and the calibrated model using inductive Venn–Abers prediction (IVAP), considering two choices of labeling tokens.

Appendix A. More metrics

For completeness, we evaluated the models using two other metrics for calibration and prediction quality: Brier loss and F_1 score (macro-averaged).

The Brier score (Brier, 1950) is the mean squared error of the N probabilistic predictions calculated on the test set:

$$L_B = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2$$

In our case, we have $y_i \in \{0, 1\}$ and p_i is the estimated probability of the positive class $P(y_i = 1)$. The Brier score loss is preferable to log loss (or cross-entropy loss) for its better handling of high-probability wrong predictions. For example, whenever $p = 0$ or $p = 1$ is returned for a wrong prediction, log loss would implode to $-\infty$. Results for Brier loss are reported in Figure 4, where we notice a similar behaviour to the ECE reported in Figure 2.

While not the ideal choice for threshold-sensitive scenarios, F_1 can simulate an "out-of-the box" setting, where the default classification threshold 0.5 is used to give binary answers. Figure 5 shows that IVAP is still the better choice, almost matched by temperature scaling for a specific choice of tokens and softmax strategy.

Appendix B. Alternative task: sentiment classification

We check the effectiveness of our approach against a different NLP task, sentiment classification. For this use case, we use the Stanford Sentiment Treebank (Socher et al., 2013), a collection of film review excerpts manually labelled with a real number $y \in [0, 1]$ represent-

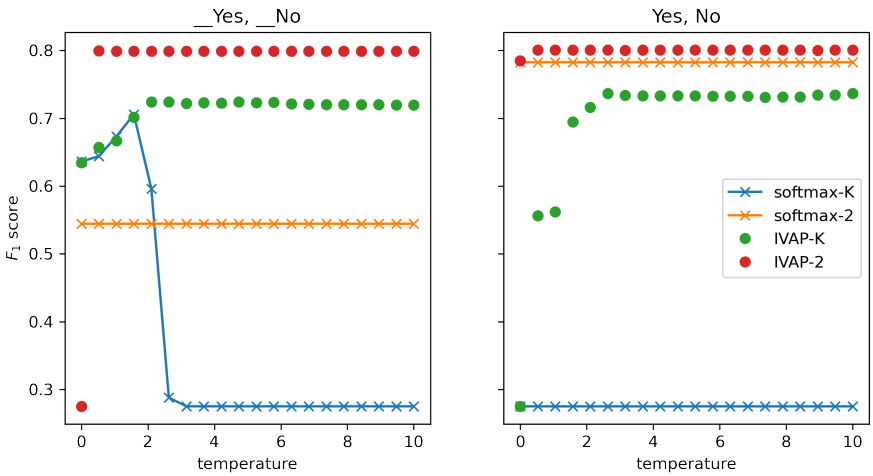


Figure 5: F_1 score for the original Llama 2 model and the calibrated model using inductive Venn–Abers prediction (IVAP), considering two choices of labeling tokens.

ing the reviewer's degree of positive sentiment. We adapt the dataset to the binary case by rounding each label to the nearest integer.

We repeat the same experiments we ran for the BoolQ dataset and find similar results, which we report here. The three default dataset splits were shuffled together and divided again in a calibration set of 2,371 examples and a test set of 9,484 examples. An example prompt is:

> Film review:
> "Enjoyably dumb, sweet, and intermittently hilarious – if you've a taste for the quirky, steal a glimpse."
> Is the review positive or negative?
> Answer:

We extract the binary answers as described in the paper, using the tokens for "Pos" and "Neg", which are present in the vocabulary unlike the tokens "Positive" and "Negative". The results are reported in Figure 6 and Figure 7, which echo the trends already noticed in the Boolean question answering case, although in this case the token choice actually makes a difference. This is likely due to the fact that there is no `Neg` token in the vocabulary, so all its scores are set to 0. The `Pos`, `_Pos` and `_Neg` tokens are instead available.

References

Ayush Agrawal, Mirac Suzgun, Lester Mackey, and Adam Kalai. Do language models know when they're hallucinating references? In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages

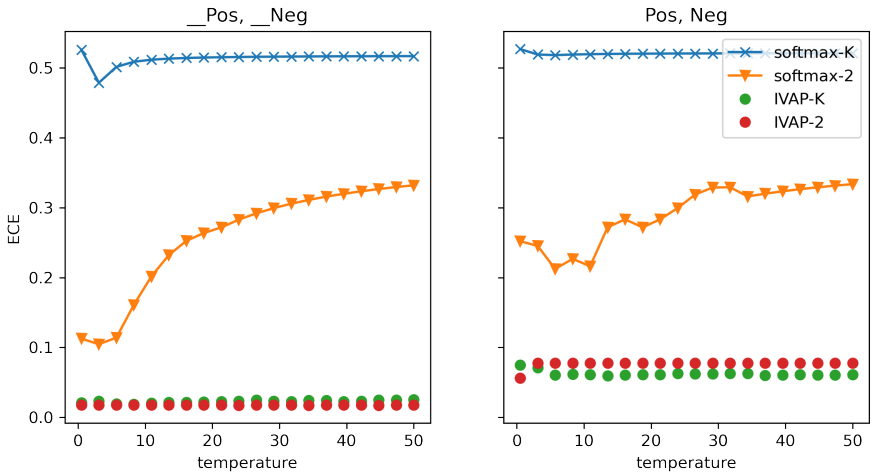


Figure 6: Sentiment classification task: calibration performance over a range of temperatures. The worse results on the right are likely due to the absence of a speficic `Neg` token.

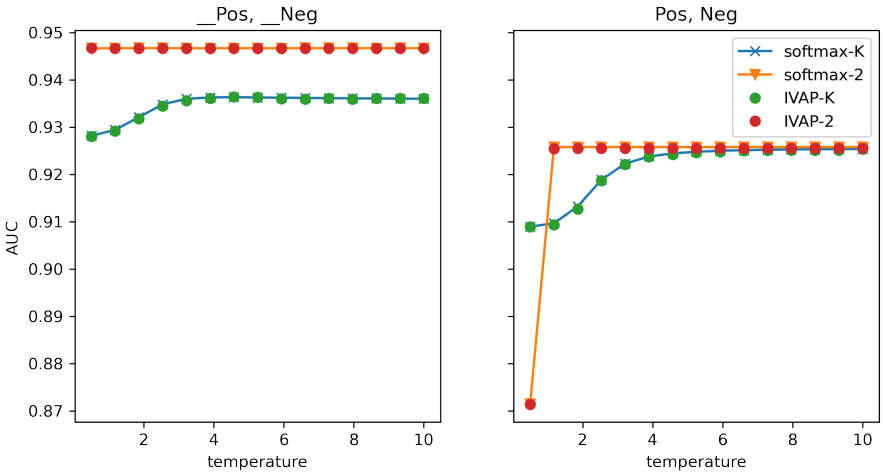


Figure 7: Sentiment classification task: prediction quality (AUC) over a range of temperatures for two labelling token choices (left and right).

912–928, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-eacl.62.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL https://aclanthology.org/2023.findings-emnlp.68.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3, 1950. doi: 10.1175/1520-0493(1950)078⟨0001:VOFEIT⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Sébastien Destercke. Uncertain data in learning: challenges and opportunities. In Ulf Johansson, Henrik Boström, Khuong An Nguyen, Zhiyuan Luo, and Lars Carlsson, editors, *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction with Applications*, volume 179 of *Proceedings of Machine Learning Research*, pages 322–332. PMLR, 24–26 Aug 2022. URL https://proceedings.mlr.press/v179/destercke22a.html.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*, 2023.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 148–155, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 155860555X.

Patrizio Giovannotti. Calibration of natural language understanding models with Venn–Abers predictors. In Ulf Johansson, Henrik Boström, Khuong An Nguyen, Zhiyuan Luo, and Lars Carlsson, editors, *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction with Applications*, volume 179 of *Proceedings of Machine Learning Research*, pages 55–71. PMLR, 24–26 Aug 2022. URL https://proceedings.mlr.press/v179/giovannotti22a.html.

Patrizio Giovannotti. Evaluating machine translation quality with conformal predictive distributions. In Harris Papadopoulos, Khuong An Nguyen, Henrik Boström, and Lars Carlsson, editors, *Proceedings of the Twelfth Symposium on Conformal and Probabilistic Prediction with Applications*, volume 204 of *Proceedings of Machine Learning Research*, pages 413–429. PMLR, 13–15 Sep 2023. URL https://proceedings.mlr.press/v204/giovannotti23a.html.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/guo17a.html.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Ulf Johansson, Tuwe Löfström, and Henrik Boström. Calibrating multi-class models. In Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin, and Khuong An Nguyen, editors, *Proceedings of the Tenth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 152 of *Proceedings of Machine Learning Research*, pages 111–130. PMLR, 08–10 Sep 2021. URL https://proceedings.mlr.press/v152/johansson21a.html.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson international edition. Pearson Prentice Hall/Pearson education international, 2009. URL http://books.google.de/books?id=crxYPgAACAAJ.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Christopher Olah, and Jared Kaplan. Language models (mostly) know what they know. *ArXiv*, abs/2207.05221, 2022. URL https://api.semanticscholar.org/CorpusID:250451161.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Arka Pal, Samuel Dooley, Micah Goldblum, and Andrew Wilson. Calibration-tuning: Teaching large language models to know what they don't know. In Raúl Vázquez, Hande Celikkanat, Dennis Ulmer, Jörg Tiedemann, Swabha Swayamdipta, Wilker Aziz, Barbara Plank, Joris Baan, and Marie-Catherine de Marneffe, editors, *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 1–14, St Julians, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.uncertainlp-1.1.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November

2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=8s8K2UZGTZ.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, page 2901. NIH Public Access, 2015.

Alec Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Shauli Ravfogel, Yoav Goldberg, and Jacob Goldberger. Conformal nucleus sampling. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 27–34, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.3. URL https://aclanthology.org/2023.findings-acl.3.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. Api is enough: Conformal prediction for large language models without logit-access, 2024. URL https://arxiv.org/abs/2403.01216.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Dennis Ulmer, Martin Gubri, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. Calibrating large language models using their generations only, 2024a.

Dennis Ulmer, Chrysoula Zerva, and Andre Martins. Non-exchangeable conformal language generation with nearest neighbors. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1909–1929, St. Julian's, Malta, March 2024b. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-eacl.129.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

V. Vovk and Ivan Petej. Venn-abers predictors. In *UAI*, 2014. URL http://alrw.net/articles/07.pdf.

Vladimir Vovk, Ivan Petej, and Valentina Fedorova. Large-scale probabilistic predictors with and without guarantees of validity. In *Advances in Neural Information Processing Systems*, pages 892–900, 2015.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer International Publishing, 2022. doi: https://doi.org/10.1007/978-3-031-06649-8.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

Chrysoula Zerva and André F. T. Martins. Conformalizing machine translation evaluation, 2023.