Toward Reliable Biodiversity Information Extraction From Large Language Models

Michael J. Elliott
ACIS Lab
University of Florida
Gainesville, Florida, USA
mielliott@ufl.edu

José A. B. Fortes
ACIS Lab
University of Florida
Gainesville, Florida, USA
fortes@ufl.edu

2024 IEEE 20th International Conference on e-Science (e-Science) | 979-8-3503-6561-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/e-Science62913.2024.10678666

***Abstract*—In this paper, we develop a method for extracting information from Large Language Models (LLMs) with associated confidence estimates. We propose that effective confidence models may be designed using a large number of uncertainty measures (i.e., variables that are only weakly predictive of - but positively correlated with - information correctness) as inputs. We trained a confidence model that uses 20 handcrafted uncertainty measures to predict GPT-4's ability to reproduce species occurrence data from iDigBio and found that, if we only consider occurrence claims that are placed in the top 30% of confidence estimates, we can increase prediction accuracy from 57% to 88% for species absence predictions and from 77% to 86% for species presence predictions. Using the same confidence model, we used GPT-4 to extract new data that extrapolates beyond the occurrence records in iDigBio and used the results to visualize geographic distributions for four individual species. More generally, this represents a novel use case for LLMs in generating credible pseudo data for applications in which high-quality curated data are unavailable or inaccessible.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The ability of Large Language Models (LLMs) to process and generate natural language text allows them to communicate information about practically anything. However, despite significant advancements in their learning capabilities [1], state-of-the-art LLMs are often found to generate information that is factually incorrect [2], [3]. This unreliability precludes the use of LLMs in practical applications that have low tolerance for factual errors. We propose that imperfect LLMs can still be useful sources of information in applications with low error tolerance if we are able to detect when LLM-generated information is correct or not. Then, LLMs may be used opportunistically whenever they are deemed trustworthy, while falling back to more traditional solutions when they are not.

We consider the information extraction problem in a biodiversity context. Specifically, we propose that information encoded within LLMs can be extracted to make predictions about the geographic distributions of individual species. To illustrate, consider the following question posed to GPT-4 [1] and its response:

> Prompt: "Can *Acer saccharum* be found in the Florida Keys? Yes or no."
>
> GPT-4: "No."

Accepting the response at face value, the LLM seems to "know" the answer to the question. Indeed, it probably does – *Acer saccharum*, more commonly known as the sugar maple, is almost certainly discussed at length in the vast text corpus on which the LLM was trained [1], and evaluations on question-answering benchmarks have demonstrated that state-of-the-art LLMs like GPT-4 are able to recall information about a wide array of topics from their training sets [1]. The ability of LLMs to recall information about the distributions of species around the world could have great utility for researchers and policymakers alike [4]. However, from the question and response text alone, there is little information to gauge how likely each response is to be correct, and thus it can be difficult for users to know when to trust LLM-generated information. In this paper, we investigate what additional sources of information are available to contextualize individual LLM-generated species occurrence predictions, and how such information can combined to gauge confidence in predictions.

Although our work is presented in a biodiversity context, our methods are more general and can be adapted to other information domains. Our primary contributions are as follows:

- We describe a novel uncertainty-based approach for extracting high-confidence information from LLMs.
- We implement the system in the context of predicting species occurrences and evaluate its performance against occurrence records accessed through iDigBio.
- We experimentally show that our confidence model can identify "high-confidence" subsets of LLM-generated species occurrence predictions to achieve a desired accuracy.
- We demonstrate the use of our methods to construct predictive occurrence distribution maps for individual species, extrapolating beyond existing occurrence data.

II. Background

A. LLMs encode information

General-purpose LLMs like GPT-4 are designed and trained to be causal language models, which predict the next word (or a short sequence of characters, called a "token") in a string of text [1]. However, there is a growing body of evidence that supports the possibility that LLMs are not merely language models which emulate natural language, but also encode factual information about the real world [5], [6]. Considering the enormous amount of text they are trained on, LLMs like GPT-4 can potentially encode any information that has been made public on the Internet. Systematic evaluations of the abilities of LLMs to provide real-world information have

The research reported in this work was funded in part by grants from the National Science Foundation (DBI 2027654) and the AT&T Foundation.

shown steady improvements in their information learning capabilities [1]. However, such evaluations only test the abilities of LLMs to answer questions to which we already know the answers. In contrast, we investigate the use of LLMs to retrieve information that is not readily available to us, where retrieving the desired information manually and in large quantities is prohibitively expensive.

B. LLM information extraction

We formalize information extraction as the process of predicting the correct answer to a well-defined query. We first distinguish between conceptual queries (hereafter, "queries") and their natural language representations. A query is a well-defined request for information, without any ambiguity. The author of the query knows precisely what sort of information is being requested and what qualifies as an acceptable answer. For our biodiversity application, we characterize queries as <species, occurrence status, location> triples, where "occurrence status" is either "present in" or "absent in". An answer to such a query should plainly indicate whether or not the triple expresses a true statement; that is, whether or not the referenced species has the specified occurrence status in the referenced location. For example, the question we posed to GPT-4 earlier ("Can *Acer saccharum* be found in the Florida Keys? Yes or no.") represents the query <"*Acer saccharum*", "present in", "the Florida Keys">, where "*Acer saccharum*" references a unique species, "the Florida Keys" references a unique location, and "present in" references a well-defined relationship between the two.

To extract information from an LLM, a query must first be translated into a format that the LLM can process. This transformation of a conceptual query into LLM-readable inputs is typically done in a two-step process: first, the query is converted into text (e.g., a natural language question, such as the one we posed earlier to GPT-4), which is then encoded as a sequence of "tokens" [7]. Both of these steps have non-unique solutions: a query may be represented as text in any number of ways, and a single sentence can have many possible token representations. The exact transformations used can be very important to the quality of extracted information, as different token representations of the same query can sometimes lead to very different outputs from LLMs [8].

Given a token sequence as input, the LLM outputs a probability distribution that expresses what tokens are likely to follow the input sequence. By running the LLM several times, the input token sequence can be extended with additional tokens. If the input sequence represent a question, then the generated tokens represent the LLM's response to the question. However, just as a natural language question is merely a representation of a well-defined query, an LLM-generated response is a text representation of an answer to the query and thus requires interpreted. The interpretation process has at least two important implications: first, if the interpreted answer is determined to be incorrect, this does not necessarily imply that the LLM did not communicate the correct answer; the fault may have been in interpretation. Second, the interpretation process is potentially lossy, meaning that the LLM's outputs may contain more information than their interpretation.

Given the sensitivity of LLMs to how their inputs are crafted and the complexity of interpreting their outputs, one should be cautious to conclude that an LLM doesn't "know" the correct answer to a query, as incorrect predictions may be induced by the use of suboptimal methods of interacting with LLMs. Consequently, when we determine that a particular prediction is likely incorrect, it may still be possible to generate better predictions from the same LLM by altering the procedure used to extract such predictions.

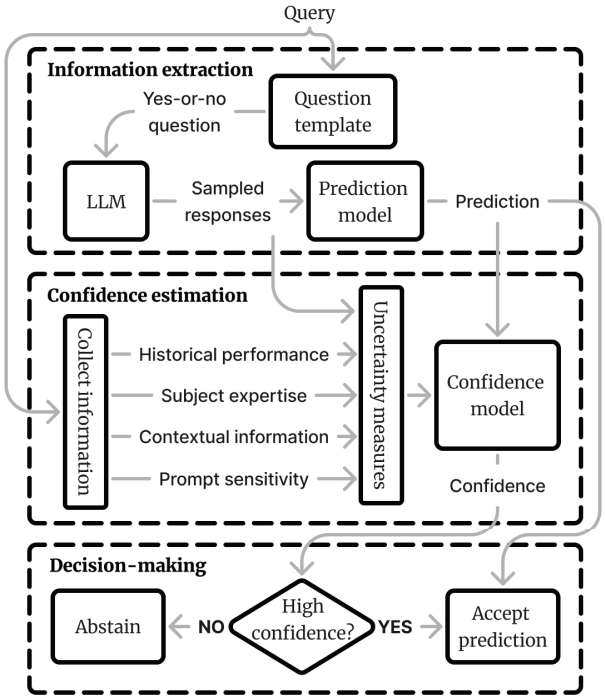


Fig. 1: Our confidence-based system for reliable information extraction consists of three logical components: information extraction, confidence estimation, and decision-making. (Top) Information extraction begins by rendering a conceptual query as a natural language question that can be used as input to an LLM. The LLM is run several times to generate a set of responses, which are are then analyzed by a prediction model to predict an answer to the query. (Middle) Confidence estimation begins by collecting information that may relate to the LLM's ability to produce a correct answer for the query. The LLM's responses are analyzed to measure intrinsic uncertainty, and supplemental data are retrieved and analyzed to measure extrinsic uncertainty. Uncertainty measurements are used by the confidence model to compute a confidence estimate for the corresponding prediction. (Bottom) The decision to either trust or discard each prediction depends on its associated confidence estimate.

III. Problem Definition

We consider the problem of extracting information from LLMs with a desired factual accuracy. This involves generating a prediction in response to each query in a query set, assigning a confidence score to each prediction, and keeping only those predictions that were assigned acceptably high confidence. The resulting high-confidence set should have greater accuracy than the full prediction set. Moreover, if confidence estimates accurately represent probabilities of correctness for the given query set, then the accuracy of high-confidence predictions should meet or exceed the desired accuracy target.

Let D denote a labeled dataset indexed by $i \in I$ and consisting of query-answer pairs $(q_i, y_i) \in D$, where q_i is a query and y_i is the correct answer (i.e., the requested information; hereafter, "the answer") to the query. For each query in the dataset, a prediction model (which makes use

of an LLM) generates a predicted answer $\hat{y}_i$ (hereafter, "the prediction") which is considered correct if $\hat{y}_i = y_i$.

For each prediction, a confidence model assigns a confidence estimate $c_i \in [0, 1]$. Each confidence estimate c_i is interpreted as an estimated probability that its corresponding prediction $\hat{y}_i$ is correct. For a fixed high-confidence threshold τ, $\hat{y}_i$ is a "high-confidence prediction" if $c_i \geq \tau$, otherwise $\hat{y}_i$ is a "low-confidence prediction".

We use the standard performance metrics of recall and precision to evaluate confidence models. For the dataset D indexed by $i \in I$, let $I_{\hat{y}=y} = \{i \in I \colon \hat{y}_i = y_i\}$ denote the set of predictions that are correct and let $I_{c\geq\tau} = \{i \in I \colon c(q_i, \hat{y}_i) \geq \tau\}$ denote the set of predictions assigned high confidence with respect to a fixed threshold τ. We formulate the precision p_c and recall r_c of high-confidence predictions as functions of τ:

$$p_c(\tau) = \frac{|I_{\hat{y}=y}||I_{c\geq\tau}|}{|I_{c\geq\tau}|} \tag{1}$$

$$p_c(\tau) = \frac{|I_{\hat{y}=y}||I_{c\geq\tau}|}{|I_{\hat{y}=y}|} \tag{2}$$

Recall (2) is the percentage of correct predictions in D that were assigned high-confidence, whereas precision (1) is the percentage of high-confidence predictions that are correct. In other words, precision is equal to the accuracy of high-confidence predictions. For both metrics, higher values indicate better confidence model performance. As a function of τ, recall can only decrease as τ increases, which leads to fewer high-confidence predictions. However, precision generally increases with τ (though in practice this is not always the case due to finite sample sizes, especially for high values of τ which yield fewer high-confidence predictions). Thus, different choices of τ trade recall for precision, and vice versa.

When $p_c(\tau) \geq \tau$ for all $\tau \in [0, 1]$, we say that the confidence model is conservatively calibrated on the test set D. Such a confidence model may underestimate – but not overestimate – the accuracy of predictions. Thus, a confidence model that is conservatively calibrated allows users to directly control the precision of a high-confidence predictions by their choice of τ.

We formulate the problem of designing confidence models for high-confidence information extraction as an optimization problem. Specifically, we wish to find a confidence model c that maximizes recall for all high-confidence thresholds τ while always remaining conservatively calibrated:

$$\begin{aligned} \max_{c} \quad & \{r_c(\tau) : \tau \in [0, 1]\} \\ \text{s.t.} \quad & \{p_c(\tau) \geq \tau : \tau \in [0, 1]\} \end{aligned} \tag{3}$$

IV. Methods

A confidence model is "useful" for enabling information extraction from LLMs with high confidence if, for a chosen high-confidence threshold, it 1) is able to output confidence estimates above the threshold, 2) does so often for a given query set (i.e., achieves acceptable recall), and 3) is conservatively calibrated for that threshold. We propose that useful confidence models may be designed that take as inputs both LLM outputs and additional variables that possess the following properties:

1) The variables incorporate information from sources other than the LLM
2) The variables are universal uncertainty measures

Property 1 – Basing confidence estimates solely on LLM outputs is problematic because the LLM can only express uncertainty that it has learned, which we call intrinsic uncertainty. Examples of intrinsic uncertainty measures include the entropy of the LLM's next-token probability assignments, the semantic entropy of its natural language text generations [9], and the detection of internal computational patterns that are associated with uncertainty [10]. For machine learning models in general, intrinsic uncertainty can be a poor indicator of prediction correctness for out-of-distribution inputs (inputs that are not well represented by the model's training data) for which the model has not learned to express uncertainty [11]. We reason that external information (anything other than the LLM's outputs for a given query) can be helpful for confidence estimation by enabling confidence models to incorporate extrinsic uncertainty which is not captured in the LLM's outputs. Confidence models whose inputs capture both intrinsic and extrinsic uncertainty can potentially detect incorrect predictions even when the LLM's outputs express low uncertainty.

Property 2 – We say that a variable is a universal uncertainty measure if it is quantitative and is maintains a monotonic relationship with prediction accuracy across test sets. That is, given two randomly selected predictions, the one with lower measured uncertainty is always at least as likely to be correct as the other. Probabilistically, this can be formulated as follows: suppose two queries are drawn at random according to probability measure P, and let q_i, y_i, and $\hat{y}_i$ denote the query, its correct answer, and a predicted answer (respectively) for two queries indexed by $i \in \{1, 2\}$. Let f be a real-valued function of each query-prediction pair, taking values $x_i = f(q_i, \hat{y}_i)$. f is a universal uncertainty measure if the following inequality holds:

$$P(\hat{y}_1 = y_1 \mid x_1 < x_2) \geq P(\hat{y}_2 = y_2 \mid x_1 < x_2) \tag{4}$$

A model whose inputs are universal uncertainty measures and enforces a monotonic relationship between its inputs and its output generates outputs that are also monotonic with prediction accuracy. That is, confidence is maximized (though not necessarily 100% confidence) when all uncertainty measures are independently minimized, and minimized when all uncertainty measures are independently maximized.

Although one uncertainty measure on its own may not have a strong enough statistical dependence with prediction accuracy to permit very high (or low) confidence estimates, we reason that a large enough collection of uncertainty measures may, as long as each uncertainty measure contributes incrementally more information about the quality of each prediction.

Our proposed approach – modeling confidence on the basis of both intrinsic and extrinsic uncertainty information, under the logical constraint that increasing uncertainty can only decrease confidence – presents the following benefits:

- Discourages overfitting – by the monotonic constraint, the confidence assigned to a single data point is bounded between the confidence estimates assigned to data points with unilaterally lower and higher uncertainties.

- Encourages rationality – the monotonic constraint prevents models from learning irrational strategies that would increase confidence in response to increased uncertainty.
- Encourages generalization – monotonic relationships between universal uncertainty measures (and by extension, confidence) and prediction accuracy can persist on novel query sets that are dissimilar to the prediction model's and confidence model's training sets.
- Easy to recalibrate – on test sets that are dissimilar to the confidence model's training set, conservative calibration may be maintained using recalibration methods like Platt scaling and isotonic regression [11].

In the subsections that follow, we describe a systematic method (Fig. 1) for extracting species occurrence predictions from LLMs (Section IV-A), measuring uncertainty in the predictions (Section IV-B), designing confidence models that process uncertainty measurements into confidence estimates (Section IV-C), and using confidence to decide when to trust extracted information (Section IV-D).

A. LLM information extraction

As discussed in Section II, we proposed to characterize species occurrence queries as <species, occurrence status, location >triples, which must first be translated into natural language text and encoded as token sequences before they can be understood by LLMs. We generate a text representation for the triple by substituting the "species" and "location" parameters into a template natural language question specific to the "occurrence status" parameter. In our experiments, we use following template for the "present in" relationship:

> "Can [species] be found in [location]? Yes or no."

For the query <"*Acer saccharum*", "present in", "the Florida Keys">, we substitute the species and location referenced in the query triple into the template and recover our original example:

> "Can *Acer saccharum* be found in the Florida Keys? Yes or no."

Noting that "absent in" is the logical opposite of "present in", we use the same template for "absent in" queries but consider "no" to be the correct answer instead of "yes."

Once a query has been converted to text, the text must then be encoded as a token sequence using a tokenizer that is compatible with the LLM to be used. For our experiments we use the OpenAI's Chat Completions API [1], which handles the token encoding process internally. In response to the token sequence representation of the query, the LLM assigns scores to all tokens in its vocabulary. Because GPT-4's token vocabulary includes "yes" and "no", the scores assigned to "yes" s_{yes} and "no" s_{no} can be used directly to make a prediction $\hat{y}$ (either "presence" or "absence") for each query. We use the following linear classifier as our prediction model,

$$\hat{y} = \begin{cases} \text{"yes"} & : \; as_{yes} + bs_{no} \geq c \\ \text{"no"} & : \; as_{yes} + bs_{no} < c \end{cases} \tag{5}$$

where model parameters a and b weight the token scores to correct for bias in the LLM toward either "yes" or "no". Because causal LLMs are trained to predict the next word in sentence, we may expect the LLM to assign high scores to "yes" and "no" even when it has no knowledge of the subject matter, simply because those are typical responses to yes-or-no questions [6]. We cannot assume it weights both choices equally.

In our use of GPT-4 via OpenAI's Chat Completions API, we could not access the model's token score outputs directly (note that this feature has since been added in recent versions of the API). Instead, we only had access to generated token sequences derived from the LLM's token score outputs. However, token scores can be approximated using sampling techniques. The task of recovering token scores is greatly simplified by our assumption that only one of two tokens will be generated – "yes" or "no" – and by only generating token sequences of length one. Specifically, we rerun the LLM n times for a given text representation of a query and count the number of "yes" and "no" responses. This process can be characterized as conducting n Bernoulli trials with a probability p that the response token will be "yes", and probability $1-p$ that it will be "no". By Bayesian inference, the unknown value of p follows a beta distribution. Denoting the number of "yes" and "no" responses as n_{yes} and n_{no}, respectively, the mean of the beta distribution is well-known:

$$\mathbb{E}[p] = \frac{n_{yes} + \alpha}{n_{yes} + n_{no} + \alpha + \beta} \tag{6}$$

where α and β define the prior distribution for p (i.e., when $n_{yes} = n_{no} = 0$). Plugging in $s_{yes} = \mathbb{E}[p]$ and $s_{no} = 1 - \mathbb{E}[p]$, the linear classifier (6) can be reformulated in terms of n_{yes} and n_{no} with new parameters a', b', and c',

$$\hat{y} = \begin{cases} \text{"yes"} & : \; a'n_{yes} + b'n_{no} \geq c' \\ \text{"no"} & : \; a'n_{yes} + b'n_{no} < c' \end{cases} \tag{7}$$

$$\begin{aligned} a' &= a - c, \\ b' &= b - c, \\ c' &= c(\alpha + \beta) - a\alpha - b\beta \end{aligned} \tag{8}$$

The system of equations (8) is linear (noting that α and β are constants) and underdetermined (there are three equations and six unknowns). Consequently, the prior parameters α and β can be ignored (they do not constrain the solutions to (8)), and a', b', and c' can be fit directly to training data. Furthermore, as linear classifiers, the optimal solution (in terms of the accuracy of $\hat{y}_i$ as a prediction for y_i) to (7) is equivalent to the optimal solution to (5) for $s_{yes} = \mathbb{E}[p]$ and $s_{no} = 1 - \mathbb{E}[p]$.

B. Uncertainty measures

This section describes the methods we used to collect uncertainty information and derive uncertainty measures. All uncertainty measures are positively oriented, i.e., greater values express greater uncertainty. Table I lists the complete set of uncertainty measures we implemented.

1) Output uncertainty: Intrinsic uncertainty measures can be derived from any set of LLM outputs, whether internal computations or the outputs of the final layer (i.e., token score assignments). Although research suggests that truthfulness information can be derived from internal computations [2], the GPT-4 API only provided access to token sequences generated from the LLM's outputs, so we only consider uncertainty measures derived from token sequences.

Given a sample of n LLM responses to a yes-or-no question, we have n_{yes} that are "yes", n_{no} that are "no", and $n_other = n - n_{yes} - n_{no}$ that were non-answers, i.e., anything other than "yes" or "no". The linear classifier's (7) prediction is $\hat{y}$. From these, we define three intrinsic uncertainty measures:

$$\begin{aligned} u_{llm,1} &= \begin{cases} n_{no}, & \hat{y} = \text{"yes"} \\ n_{yes}, & \hat{y} = \text{"no"} \end{cases} \\ u_{llm,2} &= \begin{cases} 1 - \frac{|n_{yes} - n_{no}|}{n - n_{other}}, & n_{other} < n \\ 1, & \text{otherwise} \end{cases} \\ u_{llm,3} &= n_{other} \end{aligned} \tag{9}$$

The first ($u_{llm,1}$) is the number of responses that disagree with the prediction. The second ($u_{llm,2}$) is the fraction of yes-or-no responses that agree with each other, ignoring non-answers. The third ($u_{llm,3}$) is the number of non-answers.

2) Prompt sensitivity: The outputs of LLMs can be manipulated by making seemingly superficial changes to their inputs (i.e., the "prompt") [8]. Oversensitivity to such changes may suggest that the LLM is only trying to generate responses that "sound right" rather than drawing from internalized factual knowledge. Thus, we interpret such oversensitivity as an indicator of uncertainty. To measure it, we first repeat the information extraction process for a set of differently phrased natural language questions that represent the same query (Table II). For each phrasing, we collect n responses from the LLM, count the number of "yes" and "no" responses, then predict an answer to the query using the linear classifier (7). For m different phrasings, this results in m additional predictions for each query. Let $\hat{y}_{i0}$ be the "original prediction" derived from the original phrasing (using the original question template, before rephrasing), and $\hat{y}_{ij}$ for $j \in \{1, ..., m\}$ be the predictions resulting from the m different phrasings. We define uncertainty measure $u_{ps,1}$ as the number of phrasings that resulted in predictions that were different from the original prediction:

$$u_{ps,1} = \sum_{j=1}^{m} 1[\hat{y}_{i0} \neq \hat{y}_{ij}]] \tag{10}$$

where $1[\cdot]$ is the indicator function.

Each prediction $\hat{y}_{ij}$ was derived from a score $s_j = a' n_{yes} + b' n_{no}$ calculated by the linear classifier (7). Let s_0 be the score calculated for the original prediction. We define uncertainty measure $u_{ps,2}$ as the variance of the scores that resulted from the different phrasings, including the original:

$$\begin{aligned} u_{ps,2} &= \sum_{j=0}^{m} \frac{(s_j - \bar{s})^2}{m} \\ \bar{s} &= \sum_{j=0}^{m} \frac{s_j}{m+1} \end{aligned} \tag{11}$$

where $\bar{s}$ is the average score for all phrasings, including the original.

3) Historical performance: One straightforward way to gauge confidence in an LLM's ability to provide a correct answer to a query is to consider its past performance on similar queries. Recall that we consider queries that can be represented by a specific <species, occurrence status, location> triple. Using this query structure, we can qualify how two queries are related to each other by the intersection of their "species," "occurrence status," and "location" elements. Returning to our earlier example, knowing that an LLM correctly answered the query <"*Acer saccharum*", "present in", "the Florida Keys"> could improve our confidence in the query <"*Acer saccharum*", "present in", "Miami">. Similarly, it could also improve our confidence in queries about other species being present in the Florida Keys.

To define uncertainty measures for historical performance, we first look up the LLM's accuracy on a reference set for queries with shared elements. Noting that shared "species," "occurrence status," and "location" elements can have different implications for uncertainty, we define separate uncertainty measures for each. Because higher accuracy intuitively implies lower uncertainty, we implement uncertainty measures for historical performance in terms of error rates instead of accuracy. Denoting historical accuracy for a set of related queries as acc, we define the following uncertainty measure:

$$u_{hp} = 1 - acc \tag{12}$$

Besides using shared query elements to determine that two queries are related, we also consider more indirect relationships. For example, two species may belong to the same taxonomic grouping. While such indirect relationships may be less informative of uncertainty, they have the advantage of forming larger sets of related queries. In general, the larger the query set, the more precise its historical performance measurements will be.

4) Context: LLMs like GPT-4 are causal language models, which means they are trained to predict the next word (or token) in a text segment [7]. This means that any information they communicate is learned exclusively from context (i.e., the text that precedes each token), and the more context is available for a particular topic, the more information (in terms of both precision and breadth) the LLM will "learn" about it. Because LLMs like GPT-4 are trained on trillions of tokens that cover practically everything that has been shared publicly on the Internet and more, anything that was made publicly available on the Internet prior to the LLM's training could have been learned by the LLM.

To capture uncertainty related to context, we consider the volume of context that may be available on a subject in the LLM's training set. Intuitively, the more context available related to a topic, the more the LLM could have about the topic. In our species occurrence prediction problem, where queries are characterized as <species, occurrence status, location> triples, we use the number of occurrence records available for the species identified in the query as a proxy for the amount of context available for the species. Species occurrence record counts can be collected using the online search APIs of biodiversity data aggregators like Integrated Digitized Biocollections (iDigBio). For a species with $n_{records}$ records, we define the following uncertainty measure:

$$u_{context} = \frac{1}{n_{records}} \tag{13}$$

Because we expect high record counts to be positively correlated with prediction accuracy, we take the inverse of the record count.

TABLE I: Implemented Uncertainty Measures

Type of uncertainty	Uncertainty measure	Formula	Notes
Output uncertainty	(a1) Num. matching predictions	$u_{llm,1}$	Variation in responses to repeated questions.
	(a2) Percent matching predictions	$u_{llm,2}$	
	(a3) Number of non-answers	$u_{llm,3}$	
Historical performance	(b1) Accuracy by kingdom	u_{hp}	Performance on a test set for queries with matching parameters.
	(b2) Accuracy by phylum	u_{hp}	
	(b3) Accuracy by family	u_{hp}	
	(b4) Accuracy by country	u_{hp}	
	(b5) Accuracy by state/province	u_{hp}	
Prompt sensitivity	(c1) Phrasing agreement	$u_{ps,1}$	Sensitivity of responses to 7 different question phrasings.
	(c2) Phrasing variance	$u_{ps,2}$	
Subject expertise	(d1) TaxQA accuracy for class	u_{tax}	Performance on taxonomy questions related to query parameters.
	(d2) TaxQA accuracy for order	u_{tax}	
	(d3) TaxQA accuracy for family	u_{tax}	
	(d4) TaxQA accuracy for genus	u_{tax}	
Contextual information	(e1) iDigBio records for species	$u_{context}$	Number of records in iDigBio related to query parameters.
	(e2) iDigBio records for family	$u_{context}$	
	(e3) iDigBio records for phylum	$u_{context}$	
	(e4) iDigBio records for county	$u_{context}$	
	(e5) iDigBio records for state	$u_{context}$	
	(e6) iDigBio records for country	$u_{context}$	

TABLE II: Seven Phrasings Used To Test Prompt Sensitivity

Question template
Does [species] naturally occur in [location]? Yes or no.
Can species [species] be found in [location]? Yes or no.
Is it possible to encounter species [species] in [location]? Yes or no.
Is there a presence of species [species] within [location]? Yes or no.
Does [location] harbor species [species]? Yes or no.
Is species [species] present in [location]? Yes or no.
Can one observe species [species] in [location]? Yes or no.

It may also be worthwhile to count records that are only indirectly related to the query (uncertainty measures e2,3,5,6 in Table I). For example, because *Acer saccharum* belongs to the genus *Acer* (i.e., maples), the total record count for all species of the genus *Acer* may prove useful as an additional measure of uncertainty for confidence estimation.

5) Subject expertise: Our final set of uncertainty measures concerns the LLM's expertise on the elements of the query. Whereas context-related uncertainty quantifies how much an LLM could have learned, we now directly assess what the LLM has learned. Specifically, we use the LLM's prediction accuracy on related queries that we know the answers to as a proxy for prediction accuracy on a query that we do not know the answer to.

Returning to our species occurrence example, we reason that a LLM is more likely to correctly predict <"*Acer saccharum*", "present in", "the Florida Keys"> if it is able to recite known taxonomic classifications for *Acer saccharum*. According to the Catalogue of Life (https://www.catalogueoflife.org/), *Acer saccharum* belongs to the phylum Tracheophyta. Using this information, we can construct a question to test an LLM's knowledge about *Acer saccharum*. Consider the following question posed to GPT-4, and its response:

> Prompt: "What taxonomic phylum does the species *Acer saccharum* belong to? Only say its name."
>
> GPT-4: "Magnoliophyta"

At first glance, GPT-4's answer appears to be incorrect – according to the Catalogue of Life (https://www.catalogueoflife.org/), we expected the answer to be Tracheophyta. However, species taxonomy is always evolving, and different taxonomists may use different classifications for the same species. Reviewing older taxonomic literature, we find that taxonomists actually had once placed *Acer saccharum* within a phylum called Magnoliophyta [12]. This highlights a shortcoming of LLMs – by learning from a vast corpus of text that spans many decades, LLMs are prone to reciting outdated information.

Because there may be several credible answers to questions of taxonomic classification, we consider the LLM's answer to be correct if it matches any classifications we can find. For a given taxonomic query about a particular taxonomic rank (e.g., "phylum"), let T represent a set of known taxonomic classifications, and $\hat{t}_j, j \in 1, \ldots, m$ be m responses sampled from the LLM when repeating the question m times. We define an uncertainty measure using the number of responses that do not match any known classification:

$$u_{tax} = \sum_{j=1}^{m} 1[\hat{t}_j \notin T] \tag{14}$$

C. Confidence modeling

As discussed in Section IV-A, predictions with higher measured uncertainty should never be assigned higher confidence. This is the only constraint we impose on the design of confidence models. Thus, any machine learning algorithm that can enforce monotonicity between inputs and outputs can be used to train confidence models (e.g., in our experiments in Section V, we use the XGBoost algorithm [13]).

When designing confidence models, it is important to note that monotonicity is only well-defined for two variables (e.g., one uncertainty measure and confidence estimates). In the case of multiple uncertainty measures, if one uncertainty measure is increased while another is decreased, the resulting change in confidence is unconstrained and can be either positive or

negative. However, if all uncertainty measures are increased, then by virtue of all uncertainty measures being independently monotonic (in the negative orientation) with confidence, confidence must not increase. Similarly, if all uncertainty measures are decreased, then confidence must not increase.

In our experiments, we found that the relationships between uncertainty and confidence can depend on what was predicted. Because we only consider two possible predictions, namely "yes" or "no" we found it beneficial to train two separate confidence models – one for "yes" predictions and another for "no" predictions. Then, to estimate confidence in a given prediction, we choose the confidence model that matches the prediction. Note that because we pose all queries as "present in" questions, all "yes" predictions claim species presence and all "no" predictions claim species absence for the location referenced by the query.

D. Identifying high-confidence predictions

Once a confidence estimate has been computed for a prediction, we compare the confidence estimate with a high-confidence threshold to determine whether the prediction should be trusted or not. As this work is primarily concerned with the design and evaluation of confidence models, we refrain from prescribing a method for choosing high-confidence thresholds. For our purposes, we simply note the following: for confidence model that is conservatively calibrated with respect to the set of queries it will be tested on, the expected precision achieved for any threshold is lower-bounded by the threshold itself. That is,

$$\mathbb{E}[p_c(\tau)] \geq \tau \tag{15}$$

This inequality follows directly from taking the confidence c_i assigned to each prediction $\hat{y}_i$ as a conservative estimate of the probability that the prediction is correct. Assuming the correctness of each prediction is independent of the others, the correctness of each prediction can be independently modeled as a Bernoulli trial with success probability $p_i \geq c_i$. For a set of n high-confidence predictions, let K be the number of those predictions that are correct. The expectation of K is

$$\mathbb{E}[K] = \sum_{i}^{n} p_i \geq \sum_{i}^{n} c_i \geq n\tau \tag{16}$$

Noting that $K = |I_{\hat{y}=y} \cap I_{c \geq \tau}|$ and $n = |I_{c \geq \tau}|$, the expected precision of the high-confidence predictions is

$$\mathbb{E}[p_c(\tau)] = \frac{K}{n} \geq \tau \tag{17}$$

Thus, we can expect $p_c(\tau) \geq \tau$ to hold on average, though there is always some non-zero probability that the inequality does not hold.

V. Experiment

A. Data collection

To build a reference dataset of species presence queries and answers, we first collected a sample of 12,034 species occurrence records using the iDigBio API. Each of these pairs represents the presence of a species, i.e., the correct answer to <species, "present in", location> queries represented by these pairs is always "yes". All species represented in our sample belong to either the Plantae or Animalia kingdoms. In order to meaningfully test the ability of an LLM to determine species presence, we also needed to test on pairs that represent species absence, i.e., the correct answer is "no". We generated artificial absence data by shuffling the locations in the presence dataset. The result is a set of "pseudo absences", i.e., they are not necessarily correct, but most should represent true species absences [14]. To improve the integrity of these pseudo absences, we filtered out any species-location pseudo absence pairs that matched occurrence records in iDigBio, leaving a set of 11,300 pseudo absences. Combining the presence and absence sets, we constructed a closely balanced dataset of 23,334 species occurrence records.

To test GPT-4's taxonomy expertise, we used Global Biotic Interactions' (GloBI) [15] nomer tool [16] to collect species taxonomy from many online taxonomic databases at once. The resulting taxonomy test set included classifications for 2,068 taxon names at the species, family, order, class, and phylum ranks.

To test GPT-4's understanding of individual species' geographic distributions, we constructed four additional test sets for the species *Acer saccharum*, *Amorpha canescens*, *Dasypus novemcintus*, and *Leuconotopicus albolarvatus*. Each of these test sets consists of species occurrence queries for the 3,109 counties and county-equivalents that make up the contiguous United States.

B. LLM inference

To query GPT-4, we formed natural language questions from species-location pairs using the following template: "Does [species] naturally occur in [location]? Yes or no."

When calling the OpenAI Chat Completions API, we used the "gpt-4-1106-preview" model with a top-p decoding strategy and "top_p" parameter value of 0.8. We anecdotally found that this configuration to induce enough variation in LLM responses to detect intrinsic uncertainty, while limiting most responses to either "yes" or "no" (as opposed to more creative responses that are more difficult to interpret). We collected ten single-token responses from GPT-4 per query.

C. Uncertainty measures

Table I lists the uncertainty measures we implemented, identifying the type of uncertainty each measure represents and what formula each measure follows. For the prompt sensitivity uncertainty measures, we used the seven phrasings listed in Table II. The "TaxQA" measures (d1-4) capture GPT-4's performance on the taxonomy test set. Specifically, we asked GPT-4 to classify each species in terms of the taxonomic family, order, class, phylum, and kingdom that it belongs to. We then repeated this process for each of these taxonomic ranks; e.g., we asked GPT-4 to identify which family, order, class, and phylum that each species' genus belongs to (d3), then asked similar questions for each species' family (d3), order (d2), class (d1).

D. Confidence model

To implement the confidence model, we used the XGBoost algorithm [13] as it is implemented in the scikit-learn Python package [17], with a positive monotonic constraint for each uncertainty measure. We trained two such models, one for "presence" predictions and another for "absence" predictions.

The confidence models were evaluated on the sample of iDigBio records using five-fold cross validation, in which the record set is randomly divided into five equal-sized "folds". For each fold, a new confidence model is trained on the other four folds and evaluated on the fold that was held out. This results in a set of five different performance evaluations, whose mean and standard deviation provide a more realistic characterization of model performance than when using a model and test set, especially for small high-confidence prediction sets that are sensitive to sampling noise (as in the right-hand side of the graphs in Fig. 3). The training-test folds were chosen to test the generalization ability of the confidence model. Specifically, we carefully chose the folds such that each unique species is only represented in one of the five folds. This ensures that the confidence model is not rewarded for overfitting, i.e., memorizing the training data instead of learning their underlying trends.

VI. Results

The results on the iDigBio sample test set showed that the prediction model was more reluctant to make presence predictions (predicting "yes" as the answer to a presence query) than absence predictions (predicting "no" as the answer to a presence query); while the dataset was approximately evenly split in terms of presence and absence records, our system predicted "absence" in 76% of instances.

Fig. 2 shows precision-recall curves for the two confidence models (one for "presence" predictions, one for "absence" predictions"). In both cases, higher confidence thresholds (lower recall values) correlate with higher precision. Although the overall accuracy of absence predictions is lower than that of the presence predictions, the absence confidence model produced a larger range of precision values, indicating superior performance in discriminating between correct and incorrect predictions. However, because presence prediction accuracy was much higher overall (77% accuracy compared to 57% on absence predictions), confidence on presence predictions had much less room for improvement. The precision of absence predictions only reaches the overall accuracy of presence predictions at 30% recall.

Fig. 3 shows that both confidence models stayed conservatively calibrated for thresholds below 85%, but for higher thresholds precision sometimes fell below the conservative calibration line. This is likely in large part due to the high variability of prediction accuracy for small sample sizes (i.e., at low recall values resulting from high confidence thresholds). Although the lower precision percentiles sometimes fall below the desired value for thresholds over 85%, the mean precision for both models generally exceeds the threshold.

Fig. 4 visualizes occurrence predictions for four species (one per map) as heat maps. Because we only have presence data for each species, we rely on visual comparisons with iDigBio data to evaluate prediction quality. For all four species, the prediction model generally predicts "presence" wherever there are occurrence records in iDigBio, with the notable exception of outliers (e.g., records for *Acer saccharum* in the southeast). The omission of outliers may even be advantageous in some cases, as they are often the result of species identification errors, georeferencing errors, or records for cultivated specimens that do not reflect natural species presence [18]. The results also agree with the intuition that the most uncertainty should occur at the boundaries between species presence and absence. For species with many recorded occurrences, confidence in presence predictions is highest where the occurrence data are most dense, and absence predictions are most confidence far away from the data. Additionally, our system assigned relatively high confidence to many absence predictions; this is an especially promising result, as absence data are largely missing from biodiversity datasets but important for species distribution modeling [19].

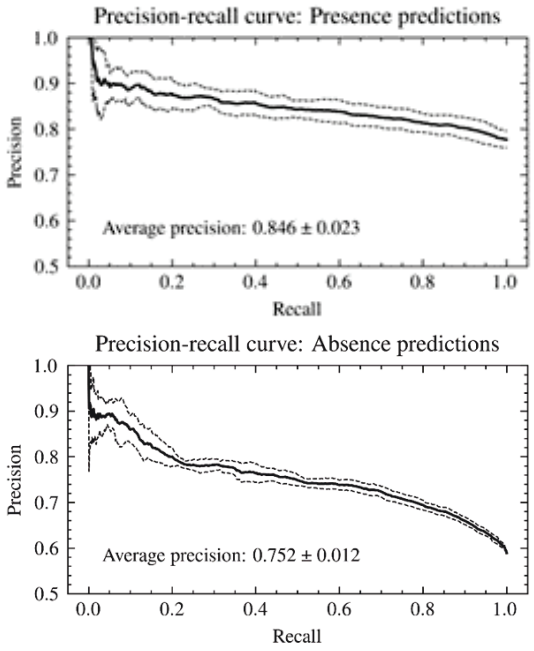


Fig. 2: Precision-recall curves for confidence estimates in GPT-4's claims of species absence (top) and species presence (bottom). Five confidence models were trained using five-fold cross validation. Solid lines represent the mean precision of the models and dotted lines represent one standard deviation from the mean.

Several of the uncertainty measures available to the confidence models are constant for predictions for the same species, and thus do not help to distinguish between correct and incorrect predictions. Reviewing the list in Table I, only 8 out of 21 measures (a1-3, a5, c1, c2, e6, and e8) vary across queries, while the remaining 13 measures (b1-4, d1-4, and e1-5) are constant. Nevertheless, the confidence model managed to make meaningful confidence distinctions for all four species.

VII. Related Work

The problem of detecting factual errors in LLM responses is often defined in terms of detecting hallucination, which refers to the case when LLM-generated text conveys information not found in its reference material (i.e., its training dataset or input text) [3]. Hallucination is not equivalent to incorrectness, though hallucinations are often assumed to be incorrect. Notably, hallucinations can sometimes be correct, and [20] even proposes that machine learning models can be trained to distinguish between correct and incorrect hallucinations. Even so, it seems reasonable that the outputs of hallucination detection models ought to be strongly correlated with LLM

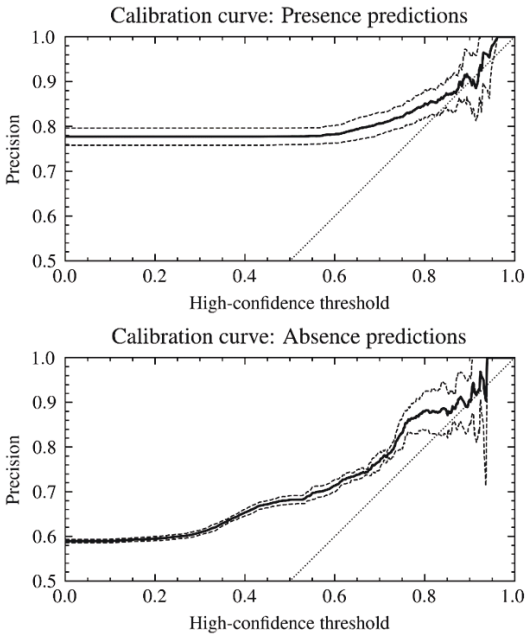
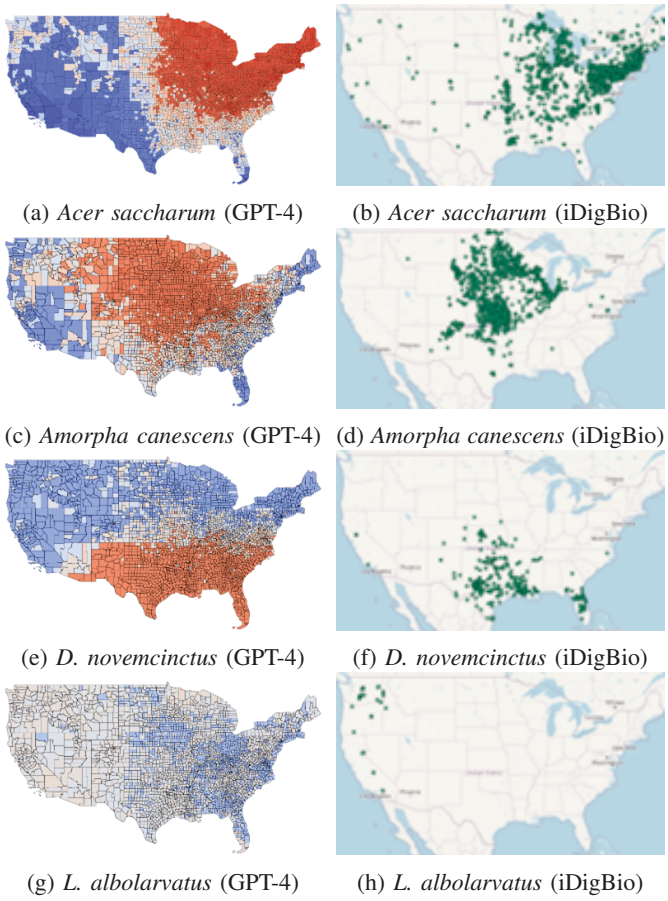


Fig. 3: Calibration curves for confidence estimates in GPT-4's claims of species absence (top) and species presence (bottom). Five confidence models were trained using five-fold cross validation. Solid lines represent the mean precision of the models and dotted lines represent one standard deviation from the mean. The straight dotted lines represent the minimum precision needed for the confidence estimates to be conservatively calibrated at each high-confidence threshold.



(a) *Acer saccharum* (GPT-4) (b) *Acer saccharum* (iDigBio)

(c) *Amorpha canescens* (GPT-4) (d) *Amorpha canescens* (iDigBio)

(e) *D. novemcinctus* (GPT-4) (f) *D. novemcinctus* (iDigBio)

(g) *L. albolarvatus* (GPT-4) (h) *L. albolarvatus* (iDigBio)

Fig. 4: GPT-4-generated species occurrence distributions and occurrence data in iDigBio across the United States for (a, b) *Acer saccharum*, (c, d) *Amorpha canescens*, (e, f) *Dasypus novemcinctus*, and (g, h) *Picoides albolarvatus*. The left column visualizes data extracted from GPT-4 and the right column shows screenshots of https://www.idigbio.org/portal/search. Red and blue colors indicate species presence and absence predictions, respectively. Confidence is visualized by color saturation, from brightly colored (high confidence) to white (low confidence).

output correctness and therefore useful for confidence estimation, especially for questions that demand more open-ended responses than "yes" or "no". [3] reviews a handful of systematic hallucination detection methods based on information extraction, question generation and answering, and comparing the text generations of different LLMs. Alternatively, [2] showed that hallucination detection (and furthermore, hallucination correction) can be achieved by training a binary classification model to recognize an LLM's internal activation patterns.

Retrieval Augmented Generation (RAG) [21] is a popular approach that has proven effective at reducing hallucination as well as increasing the informativeness of LLM responses. Although our work primarily concerns quantifying LLM response quality rather than directly improving it, RAG is similar in spirit (though certainly more sophisticated) to our incorporation of external sources of information to aid in confidence estimation. In RAG, external information is retrieved from a trusted document set, then used as context for an LLM to help inform its responses. RAG-like techniques are not intended to aid in confidence estimation, but produce various information byproducts that may prove useful as uncertainty measures, such as the scores assigned to retrieved documents. Additional uncertainty measures can also be imagined, such as quantifying the factual consistency of top-scoring documents, using natural language inference to determine the truth of a response according to retrieved documents [3], or measuring changes in text generations after applying RAG.

Our use of LLMs to predict geographic distributions of species is a novel approach to species distribution modeling (also known as environmental niche modeling), for which statistical methods (e.g., Maxent) are usually applied directly to biodiversity data and ecological data [19]. Our LLM-based approach is not intended to replace such methods, but rather should be understood as complementary. For example, high-confidence species occurrence claims generated by LLMs may serve as a more nuanced method of extrapolating occurrence data to geographical regions that are not well represented by data, but have been either studied in written reports or informal communications. Our methods can also be used to generate high-confidence pseudo absence data, which is currently far less abundant than presence data. Additionally, the reasoning abilities of LLMs may be used in species distribution modeling to incorporate information that is not easily modeled statistically, such as inferring how species behavioral traits govern their movement patterns.

VIII. Conclusions

We have shown that simple confidence models (100s of parameters) can be used to estimate confidence in information extracted from extremely complex LLMs (100s of billions to trillions of parameters) even with limited access to the LLM's outputs (only generated text responses, not the LLM's

token probability assignments or internal computations). By only trusting information that is extracted with high confidence, LLMs can be used with higher factual accuracy than is normally achieved without confidence estimation, though at the cost of discarding low-confidence information. These experimental results are further evidence of the emergent ability of LLMs to encode and recall information about the real world, beyond simply modeling natural language.

Our method relies primarily on uncertainty information to model confidence; the more uncertainty measures considered, the better the resulting confidence model will be at distinguishing between correct and incorrect information (i.e., higher confidence in correct information and lower confidence in incorrect information). By enabling the identification of high-confidence predictions, LLMs can potentially be used to generate credible pseudo data in research applications where expert-curated data are scarce or incomplete. We expect information extraction from LLMs to be of particular interest in biodiversity research, where data scarcity problems are commonplace due to the vast number of unique species have been identified and described in text but not are not yet well represented by digital occurrence data.

IX. Future Work

As an early first step toward extracting information from LLMs with high reliability, there are ample avenues for further research on this topic. One research direction is to automate the discovery and implementation of uncertainty measures. This invites deeper study into what factors determine the effectiveness of an uncertainty measure as a predictor of confidence, as well as artificial intelligence solutions to discovering and identifying relevant sources of information from which uncertainty can be quantified. There are also many promising directions for improving the overall accuracy of extracted information (not just via confidence estimation), whether by supplementing LLMs with more information via techniques like Retrieval Augmented Generation [21], allowing LLMs to refine their responses via techniques like chain-of-thought prompting [22], manipulating the internal computations of LLMs toward better factuality [2], or fine-tuning LLMs to better recall biodiversity information.

In future work, we aim to integrate confidence models into iDigBio-provided services to enable online LLM-powered biodiversity inference. To meet the high data quality standards required by research applications, we will need to extend our methods by refining the uncertainty measures that we have presented, defining new uncertainty measures, and the considering more information mediums. There are many more sources of biodiversity data and knowledge than we have considered here, such as the occurrence records served by GBIF (https://www.gbif.org/), the extensive body of biodiversity literature made accessible by the Biodiversity Heritage Library (https://www.biodiversitylibrary.org/), and the many other forms of information that are envisioned to be useful for biodiversity research as part of an extended specimen network [23].

Acknowledgment

We thank Jorrit Poelen for providing tools and guidance for the systematic collection of species taxonomy information.

References

[1] OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg, "Inference-time intervention: Eliciting truthful answers from a language model," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[3] Z. Ji et al., "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[4] I. R. Geijzendorffer et al., "Bridging the gap between biodiversity data and policy reporting needs: An essential biodiversity variables perspective," *Journal of Applied Ecology*, vol. 53, no. 5, pp. 1341–1350, 2016.

[5] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 5484–5495.

[6] Q. Wu, M. A. Khan, S. Das, V. Nanda, B. Ghosh, C. Kolling, T. Speicher, L. Bindschaedler, K. P. Gummadi, and E. Terzi, "Towards reliable latent knowledge estimation in llms: In-context learning vs. prompting based factual knowledge extraction," *arXiv preprint arXiv:2404.12957*, 2024.

[7] B. Min et al., "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.

[8] J. Gu et al., "A systematic survey of prompt engineering on vision-language foundation models," *arXiv preprint arXiv:2307.12980*, 2023.

[9] L. Kuhn, Y. Gal, and S. Farquhar, "Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation," *arXiv preprint arXiv:2302.09664*, 2023.

[10] J. Kossen, J. Han, M. Razzak, L. Schut, S. Malik, and Y. Gal, "Semantic entropy probes: Robust and cheap hallucination detection in llms," *arXiv preprint arXiv:2406.15927*, 2024.

[11] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[12] A. L. Takhtajan, "Outline of the classification of flowering plants (magnoliophyta)," *The botanical review*, vol. 46, pp. 225–359, 1980.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[14] M. Barbet-Massin, F. Jiguet, C. H. Albert, and W. Thuiller, "Selecting pseudo-absences for species distribution models: How, where and how many?" *Methods in ecology and evolution*, vol. 3, no. 2, pp. 327–338, 2012.

[15] J. H. Poelen, J. D. Simons, and C. J. Mungall, "Global biotic interactions: An open infrastructure to share and analyze species-interaction datasets," *Ecological informatics*, vol. 24, pp. 148–159, 2014.

[16] J. A. Salim and J. Poelen, "globalbioticinteractions/nomer: 0.5.6," Oct. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.10037986

[17] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[18] C. Liu, M. White, and G. Newell, "Detecting outliers in species distribution data," *Journal of Biogeography*, vol. 45, no. 1, pp. 164–176, 2018.

[19] S. M. Melo-Merino, H. Reyes-Bonilla, and A. Lira-Noriega, "Ecological niche models and species distribution models in marine environments: A literature review and spatial analysis of evidence," *Ecological Modelling*, vol. 415, p. 108837, 2020.

[20] M. Cao, Y. Dong, and J. Cheung, "Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3340–3354. [Online]. Available: https://aclanthology.org/2022.acl-long.236

[21] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[22] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[23] J. Lendemer, B. Thiers, A. K. Monfils, J. Zaspel, E. R. Ellwood, A. Bentley, K. LeVan, J. Bates, D. Jennings, D. Contreras *et al.*, "The extended specimen network: A strategy to enhance us biodiversity collections, promote research and education," *BioScience*, vol. 70, no. 1, pp. 23–30, 2020.