Quantitative Plant Biology

cambridge.org/qpb

Small language models enable rapid and accurate extraction of structured data from unstructured text: An example with plants and their specialized metabolites

Lucas Busta and Alan R. Oyler

Department of Chemistry and Biochemistry, University of Minnesota, Duluth, USA

Abstract

Transformer-based large language models are receiving considerable attention because of their ability to analyse scientific literature. Small language models (SLMs), however, also have potential in this area as they have smaller compute footprints and allow users to keep data in-house. Here, we quantitatively evaluate the ability of SLMs to: (i) score references according to project-specific relevance and (ii) extract and structuring data from unstructured sources (scientific abstracts). By comparing SLMs' outputs against those of a human on hundreds of abstracts, we found that (i) SLMs can effectively filter literature and extract structured information relatively accurately (error rates as low as 10%), but not with perfect yield (as low as 50% in some cases), (ii) that there are tradeoffs between accuracy, model size and computing requirements and (iii) that clearly written abstracts are needed to support accurate data extraction. We recommend advanced prompt engineering techniques, full-text resources and model distillation as future directions.

Original Research Article

Cite this article: L. Busta and A. R. Oyler. Small language models enable rapid and accurate extraction of structured data from unstructured text: An example with plants and their specialized metabolites. *Quantitative Plant Biology*, **6:**e26, 1–12
https://dx.doi.org/10.1017/qpb.2025.10021

Received: 30 December 2024
Revised: 2 July 2025
Accepted: 4 July 2025

Keywords:
language models; language processing; literature mining; plant chemistry; natural products.

Corresponding author: Lucas Busta;
Email: bust0037@d.umn.edu

Associate Editor:
Dr. Kirsten ten Tusscher

John Innes Centre
Unlocking Nature's Diversity

CAMBRIDGE
UNIVERSITY PRESS

1. Introduction

Language models are emerging as powerful tools for a wide array of tasks, with a particularly promising role in processing scientific literature (Agathokleous et al., 2024; Busta et al., 2024b; Jin et al., 2024; Knapp et al., 2025b; Lam et al., 2024; Simon et al., 2024). Scientific articles compile results from decades, if not centuries, of effort by scientists worldwide. However, the automation of classification, summarization and data extraction tasks related to this literature remains a challenge because natural language is a complex data type. In other fields with intricate data, such as image and sound, a proven strategy is to build mathematical models of the input data type that can then be leveraged to summarize, classify or otherwise manipulate the input. Modelling natural language is a long-standing field of study, but recently, the development and increase in accessibility of transformer-based language models have led to substantial advances in our language processing ability. Perhaps we can solve some of the many challenges with automated processing of scientific literature by applying transformer-based language models.

A considerable number of recent investigations are focused on applying large language models to scientific literature (Busta et al., 2024b; Jin et al., 2024; Knapp et al., 2025b; Sarumi & Heider, 2024; Shiu & Lehti-Shiu, 2024). For example, large language models have been utilized to perform tasks such as text classification, text summarization and question answering (Dalal et al., 2024; Guo et al., 2023; Riordan et al., 2024; Shiu & Lehti-Shiu, 2024; Yin et al., 2019). Generally, these large models require significant memory – hundreds of gigabytes – to store high billions or trillions of parameters required at runtime. However, a diverse range of language models exists beyond the popular large models from, for example, OpenAI, Anthropic, Google and Mistral. In particular, small language models (SLMs) have gained attention due to their smaller sizes (low billions or even just millions of parameters) and thus reduced computing requirements. Furthermore, though the small models are not as general purpose as the large models, the emerging evidence suggesting the small models are effective in various, albeit specific natural language processing tasks (Guo et al., 2023; Lepagnol et al., 2024; Lewis et al., 2019; Zhu et al., 2024). Thus, these small language models are intriguing because they suggest that individual scientists could use them on ordinary personal computing devices to potentially enhance scientific literature processing tasks. Importantly, running the small models on local hardware also avoids passing private and/or copyrighted content to large language model companies, which is prohibited by many research institutions and industrial organizations.

In this work, we aimed to develop and evaluate a proof-of-concept small language model processes to support the expansion of databases that document plants and the specialized metabolites that each may produce. Other databases have been created in the past to document this same type of information (Chen et al., 2017; Gallo et al., 2023; Nguyen-Vo et al., 2020; Rutz et al., 2022; Sorokina & Steinbeck, 2020; Tay et al., 2023; Xie et al., 2015; Yang et al., 2019; Zeng et al., 2024), but these databases, so far, do not leverage the potential provided by language models. We experimented with models to conduct two major tasks: (i) scoring articles based on their relevance to need-specific criteria (in this case, whether they contained reports of a specific plant making a specific chemical compound) and (ii) extracting and structuring information on the occurrence of specific chemical compounds in specific plant species. We tested a dozen language models' abilities on these tasks by manually reading, labelling and extracting data from more than 100 to more than 1000 scientific abstracts, depending on the task, then measured the models' ability to perform those same tasks. Overall, our findings indicate that small language models, while not perfect, effectively aid in filtering scientific literature references and in extracting data. We recommend that researchers both experiment with these models and monitor for updates in literature processing software that incorporate language model-enabled features.

2. Results and discussion

To develop and evaluate a potential role for small language models in creating a phytochemical occurrence database, we assessed such models' abilities with regard to two tasks: (i) to quickly score references according to whether the reference reports the occurrence of a specific compound in a specific plant species (Task 1, Section 2.1) and (ii) to evaluate language models' ability to extract an experimentally supported compound occurrence dataset (Task 2, Section 2.2). For these investigations, we chose to use six triterpenoid compounds as test cases (Figure 1a). The six triterpenoid test cases under study here presented a challenge

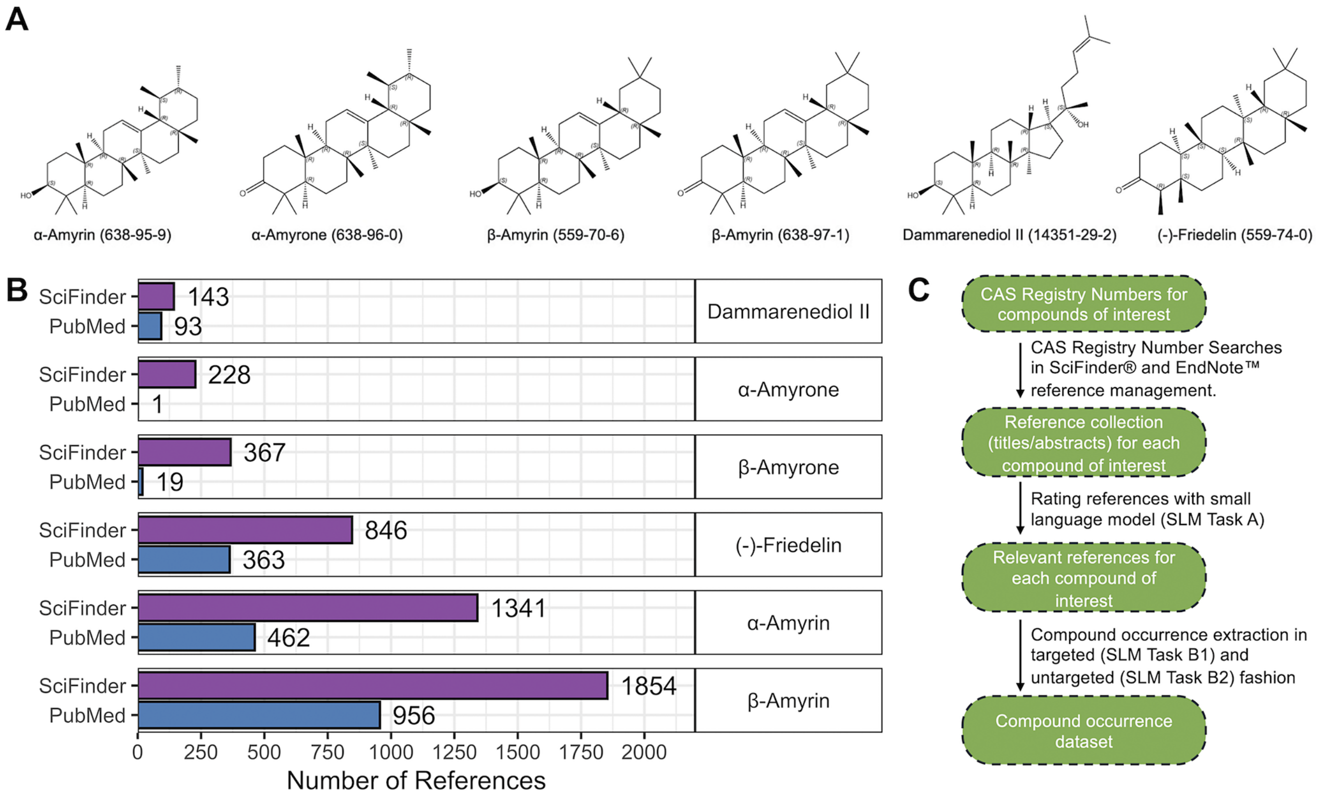

Figure 1. Comparison of SciFinder® versus PubMed® as a data source and schematic of the small language model workflow for retrieving compound–species associations from literature. (a). Structures, common names and CAS Registry® numbers for the six triterpenoid compounds used as test cases in our small language model development and evaluation work. (b) Bar plot comparing the number of references (*x*-axis) found by SciFinder®and PubMed® (*y*-axis) for the six different triterpenoids (vertically arranged panels) studied in this work. Each bar represents the number of references found by the indicated search tool for a particular triterpenoid. The absolute number of references found is shown in text to the right of each bar. Bars are colour coded according to search tool (SciFinder® in purple and PubMed® in blue). SciFinder® searches were conducted using CAS Registry® numbers, while PubMed® (which does not generally use these registry numbers) searches were conducted using compound common names. (c) Schematic for the workflow we developed to extract compound occurrence data from information in the literature. Files or information are shown in green bubbles, while steps or actions are shown as arrows. The workflow consists of searching the literature with SciFinder® based on CAS Registry® numbers then creating a repository of references and associated full-text PDF files in an EndNote$^{\text{TM}}$ database; then filtering references for those of highest task-specific relevance (SLM Task A) and finally extracting compound occurrence data in either a targeted (SLM Task B1) or untargeted (SLM Task B2) fashion. Abbreviations: SLM: small language model.

because they have been mentioned in the literature (going back to the 1960s) by many names. Indeed, CAS SciFinder® indicates that a total of more than 52 names has been associated with these six compounds, potentially complicating efforts to retrieve references describing the occurrence of specific plant chemicals. Fortunately, triterpenoids (and the vast majority of all other chemical entities) are identified explicitly by their CAS Registry® numbers (Figure 1a), which means that references to a given compound that use varied nomenclature can be collected simultaneously and non-ambiguously when using CAS Registry® number-based search strategies. While other identification number systems exist, such as PubChem® and LOTUS numbers, these alternate systems are not as comprehensive as CAS Registry® numbers. Thus, where possible, searches with identification numbers, as opposed to common names, are preferred because this approach ensures not only that a broader array of references is retrieved, but also that those reference relate to one and the same compound, including the correct stereochemistry.

To obtain references describing our six triterpenoids of interest, we used CAS Registry® numbers to search SciFinder®, which, though requiring a subscription, allows the user to enter a CAS Registry® number and then navigate directly to literature references that relate to that specified compound. PubMed®, though providing open access, does not generally support searches based on CAS Registry® numbers or PubChem ID numbers, so we conducted searches in PubMed using compound common names. We first considered the two most common compounds in our case study set, α- and β-amyrin. In SciFinder®, we found over 1340 and more than 1850 hits for these two compounds, respectively, compared to fewer than 500 and 1000 hits in PubMed® (Figure 1b). Results were similar for the other four triterpenoid test cases (Figure 1b). In total, ~3200 SciFinder® references were retrieved using our searches, while ~1500 references were retrieved by PubMed®. Therefore, we used SciFinder®-retrieved references to develop and evaluate small language model-based reference ranking and occurrence dataset extraction processes (Figure 1c).

2.1. SLM Task A: Rating references according to relevance with a small language model

At this stage in this work, we had used SciFinder® to collect more than 3000 references associated with one or more of the six triterpenoids that comprised our test cases for compound occurrence data collection. Our first aim was to determine the efficacy of small language models with respect to filtering the references for articles of interest. In this case, our interest was in articles that reported phytochemical occurrences (i.e. evidence for a specific plant species producing a specific chemical compound). To establish a benchmark against which to evaluate small language model performance, we read more than 1500 of the references in our collection, including their titles and abstracts and classified each as ‘reporting an occurrence’, ‘maybe reporting an occurrence’ or ‘not reporting an occurrence’ (Supplementary Materials S1). These human-read citations included all the reference citations for α-amyrone, β-amyrone, dammarenediol II as well as (−)-friedelin. For an article to be considered as ‘reporting an occurrence’, its title or abstract needed to indicate that the article in question provided experimental evidence for the presence of a particular plant chemical in a particular plant species. Articles whose titles or abstracts merely contained co-occurrences of a plant chemical name and a plant species name without indicating that there was experimental evidence for an association between the two were classified as ‘not reporting an occurrence’. Citations that did not explicitly indicate that their articles contained experimental evidence for a compound’s occurrence but instead implied that such evidence might be present in the full text (to which we did not have access) were classified as ‘maybe reporting an occurrence’. Of the 1558 references that we read, 720 were classified as ‘reporting an occurrence’ (46%), 332 were classified as ‘maybe reporting an occurrence’ (21%) and 506 were classified as ‘not reporting an occurrence’ (33%).

We next evaluated how well language models could classify references according to whether they reported the occurrence of a phytochemical using the 1558 manually classified references as a ground-truth set. We used the bart-large-mnli model, selected because it is one of the most downloaded on Huggingface.co, a major hub for open-source language development, largely due to its versatility and high speed – we found that it could process 45,000 articles/hour, a desirable characteristic for a model that will be used to filter inputs into a multi-step processing pipeline. This small language model is employed by providing it with a body of text and then one or more classifier phrases. The model then assigns a score to each phrase to indicate how closely that phrase relates to the provided text. The bart-large-mnli model card (i.e. the instruction manual) suggests presenting the model with a classifier phrase framed as a hypothesis (e.g. ‘This text is about politics’). Accordingly, we investigated phrases such as ‘Amyrin is present in plants’ as well as paired phrases in which a hypothesis was matched with the exact negative (i.e. ‘Amyrin is present in plants’ and ‘Amyrin is not present in plants’). Our early experiments showed that composite scores derived from the pairs’ individual scores improve the signal-to-noise ratio in the classification task. Furthermore, we noted that multiple compound names could be included in these positive and negative phrases (for instance, ‘friedelin, friedooleanan-3-one, friedelan-3-one, friedelanone or friedeline is found in plants’; full-classifier phrase details are provided in Supplementary Materials S2). In future, large-scale operations, a single, general classifier phrase, which is not based on compound names, would be preferred if the performance was comparable to that of our specific classifier phrase system, which is based on compound names. Therefore, we also tested the more general classifier phrase, ‘the text discusses plants that contain specific compounds’.

Using the two classifier phrase approaches described in the previous paragraph, we instructed the bart-large-mnli model to assign two scores to each of the 1558 references, one composite score from the binary/two classifier phrase system, as well as a score for the general classifier phrase. Composite scores (means and standard deviations) for, respectively, references that reported occurrences/maybe reported occurrences/did not report occurrences for (−)-friedelin were 0.9 ± 0.1, 0.8 ± 0.1 and 0.7 ± 0.1 (Figure 2a, top panel). Results were similar for the other three triterpenoids (Figure 2a). Scores from the general classifier phrase for, respectively, references that reported occurrences/maybe reported occurrences/did not report occurrences for (−)-friedelin were 0.9 ± 0.06, 0.9 ± 0.04 and 0.8 ± 0.2 (Figure 2b, top panel) and again, results were similar for the other three triterpenoids (Figure 2b). This illustrates that the two-classifier phrase system and the general classifier phrase system both worked comparably well among references describing four different triterpenoid compounds and may also to a similar extent for compounds other than triterpenoids.

Next, we investigated the ability of these scores to act as a filter to separate articles of interest that report chemical occurrences from those that did not report such occurrences. Thus, we examined

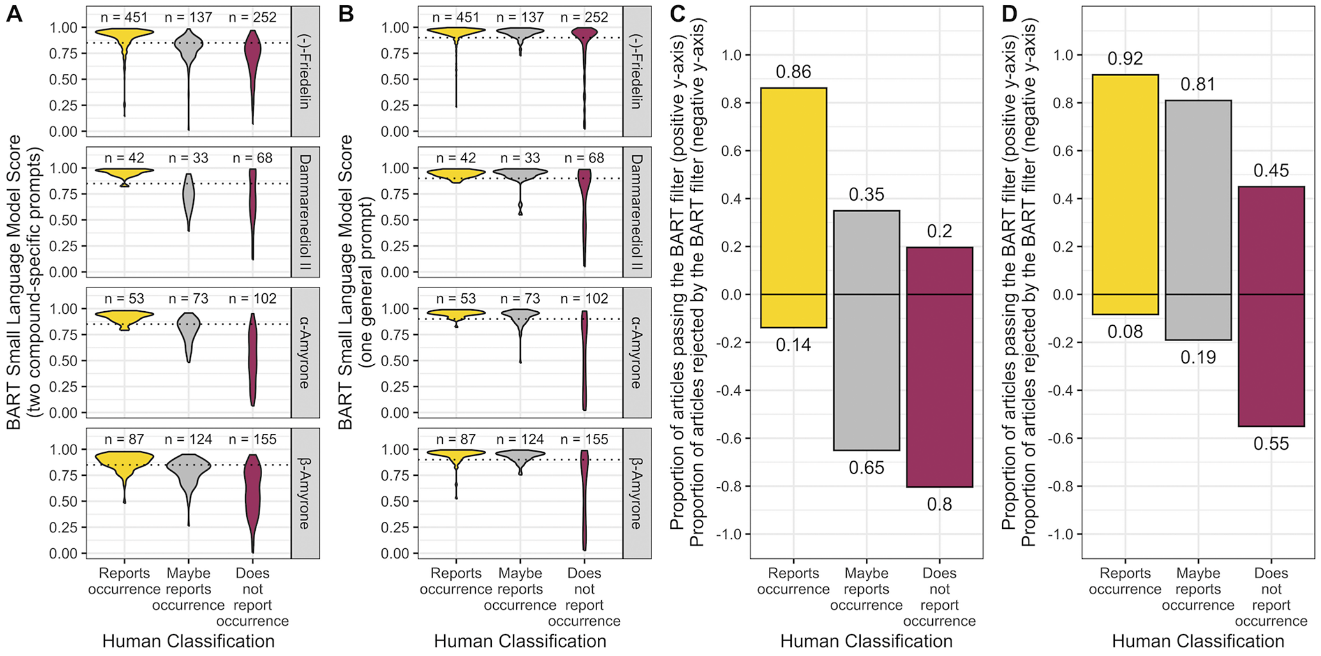

Figure 2. Performance of small language models on a reference relevance ranking task. (a, b) Violin plot showing the score (BART small model language score, *y*-axis) assigned to references by the bart-large-mnli small language model. Scores range from zero (low relevance) to one (high relevance) and indicate the relevance of a given reference to a user-defined natural language criterion. In panel a, the score is derived from two, chemical compound-specific criteria (full details in methods section), while in panel b, the score is derived from a single, generic criterion ('chemical compounds are found in plants'). In both panels, scores are broken out according to whether the reference was labelled by a human as 'reporting an occurrence', 'maybe reporting an occurrence', 'not reporting an occurrence' of a specific chemical compound in a specific species (*x*-axis). The number of references belonging to each group are shown above each violin. In panel a, the dotted line represents a threshold of 0.85, and in panel b, the dotted line represents a threshold of 0.9 (details of thresholds discussed in main text). (c, d) Column plot showing the proportion of references (*y*-axis) from each human labelled category ('reporting an occurrence', 'maybe reporting an occurrence' or 'not reporting an occurrence'; *x*-axis) that would be retained if a threshold small language model score was used for filtering references. The proportion of each column in the positive *y* space indicates the fraction of references that would pass the filter and be retained, while the proportion of each column in the negative *y* space indicates the fraction of references that would be rejected by the filter and eliminated. Exact proportions are shown in numbers above and below each column. In panel c the threshold is 0.85, based on two-prompt scoring, while in panel d the threshold is 0.9, based on single, general prompt scoring (details in main text and methods section). For example, if a score of 0.85 were used as a threshold with which to filter references that had been scored using the two-prompt small language model scoring system, then 86% of references reporting occurrences would be retained while 14% of such references would be rejected, 35% of references maybe reporting occurrences would be retained while 65% of such references would be rejected and 20% of references not reporting occurrences would be retained while 80% of such references would be rejected. In all panels a–d, colours correspond to the three human label categories ('reporting an occurrence', 'maybe reporting an occurrence', 'not reporting an occurrence'). BART stands for the bart-large-mnli small language model.

the proportion of the former type articles that would be retained if a threshold score were to be used as a filtering criterion for the reference collection (i.e. if references with a score higher than a threshold were to be retained and those with a score lower than the threshold were to be eliminated from the collection). Based on the distribution of scores assigned to articles that reported chemical occurrences versus those that did not (Figure 2a,b), we selected 0.85 as a threshold for the specific two-prompt scores and 0.90 as a threshold for the general prompt-derived scores. With these thresholds, the specific two-prompt scoring system acting as a filter would have retained 86% of the references that report phytochemical occurrences (the references of interest in our study) and rejected 80% of the references that did not report an occurrence (Figure 2c). The general prompting system, with a 0.90 filtering threshold, would have retained 92% of the references reporting phytochemical occurrences and eliminated 55% of the references that did not report occurrences (Figure 2d). While both the two-prompt and general prompt filtering approach led to the retention and rejection, respectively, of article of interest and not of interest, the two approaches handled articles we had labelled as 'maybe reports an occurrence' differently: the two-prompt approach kept only 35% of these, while the general approach kept 81%. To learn more about these 'maybe' references, we obtained and read 100 full-text articles for these references (those related to α-amyrone and dammarenediol II, Supplementary Materials S3). This manual inspection revealed that approximately 65% of these 'maybe' references contained reports of compound occurrence data, which suggested that access to full-text information will help create more comprehensive chemical occurrence datasets. After manual re-annotation of the 100 articles based on full texts, we tested to see if the scores of occurrence-reporting articles differed from articles that did not report occurrences, but there was no significant difference in the scores. However, regardless of whether full texts are available or not, our results show that small language model relevance scores provide a means to quickly (~45,000 references/hour) and accurately (~80% relevant articles kept, ~80% of irrelevant articles rejected) identify references that are most likely to provide the information that a user might be seeking. This ability will be highly useful when dealing with many thousands of references. Our data also indicate that there will likely be a benefit to developing more nuanced filtering approaches to handle edge cases like the 'maybe' articles we identified here.

2.2. SLM Task B: Extracting compound occurrence data with language models

After filtering our collection of references to include only entries with high scores concerning phytochemical occurrence data, we evaluated the ability of language models to extract experimentally supported compound presence details. In this task, two steps can be envisioned: (i) a first step in which a model receives a body of text including the title and abstract of a scientific article and (ii) a second step in which a model receives a query about compound occurrences. For example, in the second step, we might ask the model: 'Does the provided text offer experimental evidence that *Arabidopsis thaliana* produces the chemical compound thalaniol?' This mode of operation represents a *targeted approach*. A second mode of operation (for the second step) could be to pass a language model a text passage containing the title and abstract of a scientific article and pose an open-ended query such as: 'List all of the plant species mentioned in the provided text and indicate which chemical compounds were reported from each one as part of the experimental investigation described in the passage'. This second mode represents an *untargeted approach*. Several advantages and disadvantages of each approach can be imagined from the outset. For example, an untargeted approach does not require a preconceived set of chemical compounds or plant species of interest about which to query the model, and a single untargeted query can potentially extract multiple compound-occurrence data simultaneously. In contrast, one benefit of the targeted approach is the relative simplicity of creating human-labelled data. Thus, a true/false answer about one plant/compound occurrence can be supplied by the human or model instead of meticulously generating a complete list of such occurrences. Furthermore, a model's rate of detecting true negative associations can be measured directly by comparing the model's response to plant and compound names appearing in an abstract, without experimental association data, to the corresponding human response. Thus, the targeted and untargeted approaches each offer distinct benefits, so we tested and herein present results from both approaches. For either approach, a model must correctly distinguish between characters used in chemical names (in this study, especially Greek letters like α and β) and recognize the synonymous nature of certain symbols and words (e.g. that α-amyrin and alpha-amyrin are the same compound). Previous studies have shown that Greek letters occupy their own positions in language model input spaces (Stevenson et al., 2024) and that such models can reason over diverse alphabets (Maronikolakis et al., 2021), suggesting that modern language models, in this regard at least, may be suited to the earlier described approaches. We conducted preliminary tests by asking each model (see model details next) a series of six questions like 'are α-amyrin and alpha-amyrin the same compound?', 'are α-amyrin and beta-amyrin the same compound', 'are β-amyrin and beta-amyrin the same compound' and so forth. All but the smallest two models (gemma-3-1B-instruct and qwen-2.5-0.5B-instruct) answered these questions with 100% accuracy, illustrating that detailed investigations of task/project-specific assumptions should be empirically tested during the model selection step of a language model-based investigation.

2.2.1. SLM Task B1: Targeted compound occurrence data extraction.

To evaluate the ability of large language models to extract compound occurrence data from scientific abstracts, we first prepared and manually evaluated a set of candidate occurrences. For this effort, we used regular expression-based pattern matching to identify accepted plant species names in the abstracts associated with the six triterpenoids that comprised the present test case. We then compiled a dataset containing three columns: the title and abstract of each reference, the chemical compound linked to it (the SciFinder® search compound that retrieved that reference in the first place) and accepted plant species name(s) found in that title or abstract. We manually evaluated 500 candidate associations and annotated each occurrence as positive (the abstract described experimental support for the occurrence of that compound in that plant species) or a negative (the abstract did not provide such support). We found that roughly 350 (71%) of the candidate associations were negatives, while around 150 (29%) were positives (Supplementary Materials S4). With a set of human-labelled compound species or candidate compound species associations in hand, we next turned to evaluating whether open-source language models could perform the same task. For this task, we used open-source language models that accepted two types of prompts. The first prompt was a system prompt that contained detailed instructions on how the model should generate an output. The second prompt (also called user text) delivered content from which the model generated that output. We used the second prompt to supply information on the candidate compound species association (title/abstract, compound name and species name) and the system prompt to convey detailed instructions on how the model was supposed to evaluate this given information (full details in Methods).

Past research has shown that language models of different sizes vary in their ability to perform natural language processing tasks (Brown et al., 2020; Kaplan et al., 2020), including tasks related to chemical occurrence data extraction (Busta et al., 2024a). Accordingly, in evaluating their capacity for the present targeted occurrence extraction task, we tested 12 language models of various scales, spanning 0.5 billion to 32 billion parameters (often denoted 0.5B to 32B, Figure 3a). These models included variants of different sizes from the Qwen family (Qwen: An et al., 2025) (32B, 14B, 7B and 0.5B), the Gemma family (Gemma et al., 2025) (27B, 12B, 4B and 1B) and the Phi-4 family (phi-4 14B and phi4-mini-instruct 4B) (Abdin et al., 2024). Each model was given the same system prompt and all 500 candidate occurrences that had been previously examined manually. During these assessments, all models were run at 16-bit precision, except gemma-3-27B-it-unsloth and phi-4-unsloth-bnb-4bit, which are dynamically quantized instances operating at 4-bit precision (Figure 3a). When reviewing the 500 candidate associations, run times generally varied in direct proportion with size; qwen-2.5-32B-instruct handled about 200 references per hour, while qwen-2.5-0.5B-instruct surpassed 32,000 per hour (Figure 3a). Notably, the quantized variants processed references at speeds only slightly higher than their full-resolution counterparts (e.g. the 4-bit phi-4-unsloth at 1500 references/hour and the 16-bit phi-4 at 1200 per hour). These speeds will be important when applying language model-based approaches to larger projects or the assembly of databases.

Alongside measuring how quickly various models processed 500 candidate associations, we also examined model accuracy. To gauge that accuracy, we compared whether each model labelled every candidate association as positive or negative against the corresponding human label. The results let us classify each model output as a true positive (when the model labelled a candidate association as positive, matching the human label), a true negative (when both the model and the human labelled it negative), a false positive (when the model labelled it positive, but the human did not) or a false negative (when the model labelled it negative, but the human did not). Because 71% of the 500 candidate associations were negative, a high-performing model would have a true negative rate approaching 71%. The true negative rates for the models tested ranged from 52% to 67%, with models containing more parameters generally showing higher percentages (Figure 3b). One exception was qwen-2.5-0.5B-instruct, which had a 0% true negative rate, as it labelled all candidates occurrences as positive. These

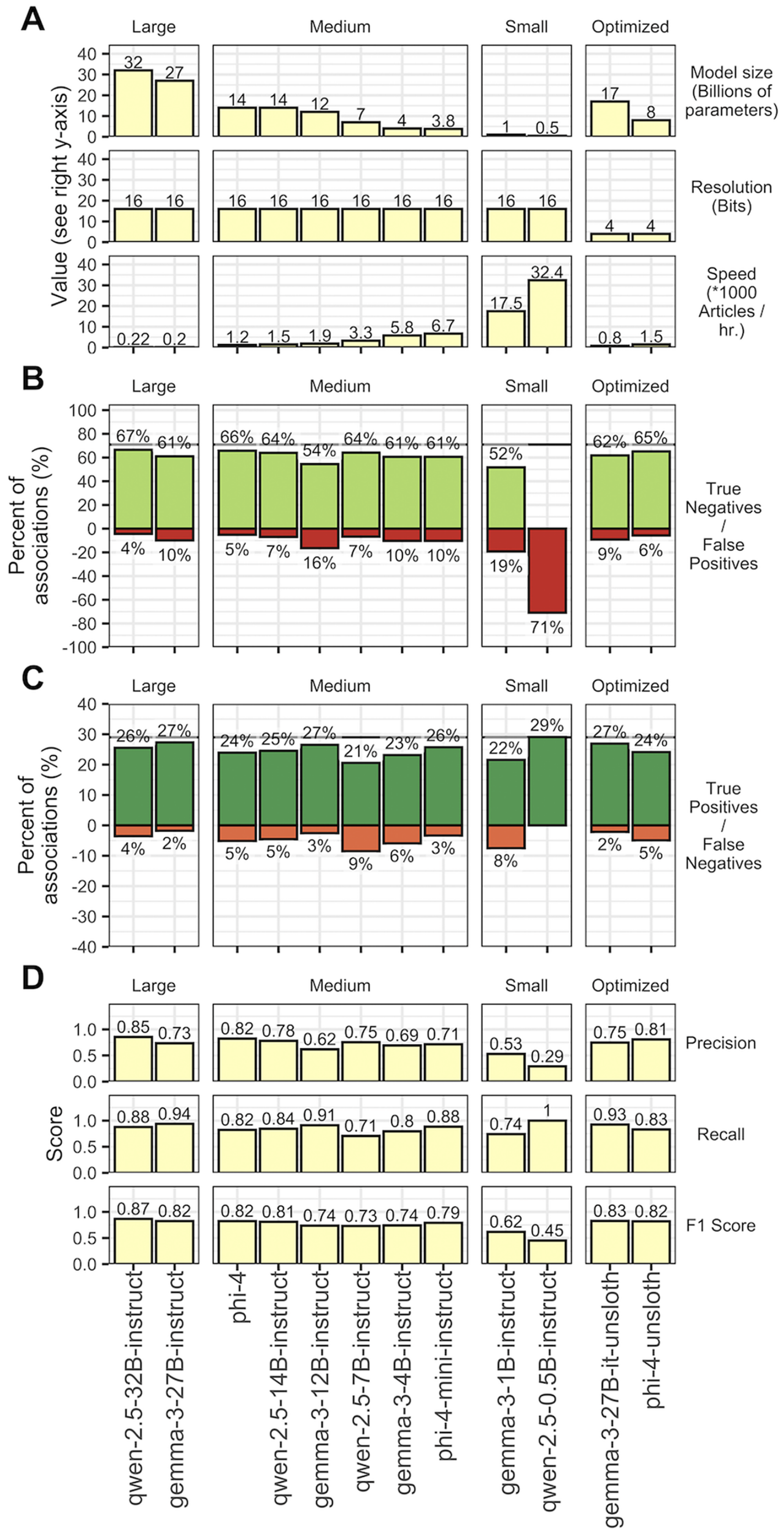

Figure 3. Performance of language models on a targeted compound occurrence data extraction task. (a) Bar plot showing various metrics (*y*-axis in each row of panels) for different language models (*x*-axis). The first row shows model size in billions of parameters, the second row shows model resolution in bits, the third row shows the speed with which a model processes references (using the prompt shown in the methods section) in units of 1000 references per hour. (b) Bar plot showing the raw performance metrics of each model (false negative, false positive, true negative and true positive rates). False negatives arise when a model erroneously marks a real compound occurrence as not being true. False positives arise when a model erroneously marks a simple textual occurrence of a compound name and species name as an occurrence data point. True negatives arise when a model correctly marks a simple textual occurrence of a compound name and species name as such and not as an occurrence data point. True positives arise when a model correctly marks a compound occurrence as such. According to human evaluation of the 500 putative occurrences used to test the models, 71% of the putative occurrences were real (i.e. 'positives') and 29% of the putative occurrences were just textual co-occurrence (i.e. 'negatives'). Thus, a perfect model would have, in this experiment, a 71% true negative rate and a 29% true positive rate. Bars are coloured according to true/false positive/negative. (c) Bar plot showing the processed performance metrics of each model. In the first row, the precision of each model is shown (the ratio of true positives to the sum of true positives and false positives). In the second row, the recall of each model is shown (the ratio of true positives to the sum of true positives and false negatives). In the third row, the F1 score is shown, which is the harmonic mean of the precision and recall. In (a–c), models are organized into columns of panels by type (large: > 20 B parameters, medium: 1–20 B parameters, small: 0–1 B parameters and optimized: 4-bit resolution models).

differences in true negative rates came with parallel differences in false positive rates, since false positives arise when a model incorrectly labels a negative result as positive. The false positive rate is one of the most important metrics for this task because those errors represent fabricated occurrence data. In our experiments, larger models achieved lower false positive rates overall, with qwen-2.5-32B-instruct and phi-4 showing the lowest values at 4% and 5% respectively (Figure 3b). Because both were also the slowest and largest, there is a clear trade-off between parameter count and computational requirements on one hand and task-specific accuracy on the other.

The models we tested here did not only vary in their (true negative)/(false positive) rates, but also in their (true positive)/(false negative) rates. Since positive associations comprised 29% of the 500 candidate associations, a perfect model in our experiment would have a 29% true positive rate. True positive rates among the models tested here generally ranged from 21% to 27% (Figure 3b). This variability did not correlate as strongly with model size as did the (true negative)/(false positive) rates. For example, a large model (qwen-2.5-32B-intruct), two medium models (gemma-3-12B-instruct and phi-4-mini-instruct (4B)) and one of the quantized models (gemma-3-27B-it-unsloth) all had very similar true positive rates (26% or 27%, Figure 3b). Note that the perfect true positive rate of qwen-2.5-0.5B-instruct is a misleading statistic, since this model simply labelled all associations with which it was presented as positive. To account for such potentially misleading rates, we computed precision and recall statistics. Precision is calculated as the number of true positive results divided by the sum of true positive and false positive results, which indicates how reliable the model is when it marks an association as positive. Recall is calculated as the number of true positive results divided by the sum of true positive and false negative results, which reflects the model's ability to correctly identify all actual positive associations. Excluding qwen-2.5-0.5B-instruct, precision varied from 0.5 to as high as 0.85 and recall varied from 0.71 to 0.94 (Figure 3c). We also computed F1 scores, which are the harmonic mean of precision and recall, to provide a single metric to balance both reliability (precision) and completeness (recall). F1 scores (excluding qwen-2.5-0.5B-instruct) ranged from 0.62 (gemma-3-1B-instruct) to 0.87 (qwen-2.5-32B-instruct) and varied, again, according to model size, which reinforced the importance of that parameter in task-specific accuracy.

So far, our results indicated that language models can assess whether an abstract describes experimental support for a particular compound, but no model was entirely accurate in performing this task. Accordingly, we next turned our attention to a detailed examination of the candidate associations that were frequently labelled incorrectly by the language models. Specifically, we reviewed the incorrect answers generated by the phi-4 model. First, we focused on references in which no experimental support for a compound's occurrence was provided, yet the model (erroneously) indicated such support was presented (i.e. false positives). Among these occurrences, two main text structures appeared to 'confuse' the model. The first scenario involved abstracts where occurrence data were not presented in separate sentences but instead merged with multiple data types. For example, some passages combined information from authentic standards and plant extracts or from sediments and plant extracts or listed multiple compounds from several species in a single statement. The second scenario leading to false positives involved abstracts that failed to provide clear statements about plant/compound occurrences, even to a human reader. As an example, one such abstract stated 'beta-sitosterol and alpha-amyrin were isolated from unsaponifiable fractions of mature seeds of solanaceae plants' and mentioned the solanaceous species *Hyoscyamus muticus*, which caused the model to label alpha-amyrin as present in *H. muticus*, even though this link was not explicitly supported by the text. Finally, we examined references where positive associations were mistakenly labelled by the models as negative (i.e. false negatives). We identified three main cases: (i) abstracts that were written in confusing ways, which lead the model to produce an incorrect result, (ii) abstracts that contained an alternative spelling or abbreviation for a compound or species name and (iii) clearly written abstracts in which the model nevertheless failed to provide the correct answer. These scenarios appeared in roughly equal proportions among phi-4's false negatives. To summarize, the model sometimes makes clear mistakes, but, just as often, the model produces incorrect answers because of inconsistencies or unclear information in the input data. Finally, we also examined the performance of the models when alternative spellings of compound names were present in abstracts. Across the 500 candidate associations we manually evaluated, there were 28 instances where alternative spellings were used in the abstract (amyrin/amirine, friedelin/friedeline, amyrone/amyrenone). Evaluating these candidate associations, the highest performing models were correct ~50% of the time, which is lower than model performance across the entire dataset (~10% overall error rate). Thus, we conclude that these alternative spellings do impact model performance and strategies to deal with such should be included in the design of small language model-based pipelines.

Several conclusions arise from our work with targeted compound-occurrence dataset extraction. First, models with more parameters ('larger' models) appear to perform the task with higher accuracy, though that improvement comes alongside increased computational demands and time requirements. To balance speed and performance, architectures such as phi-4 stand out from those evaluated in this study. Next, the abilities of systems like phi-4 to accurately detect true negatives indicate that they are distinguishing references with textual co-occurrence of plant and compound names from references that present experimental evidence for a plant producing a given compound. Finally, our examination of the underlying reasons for incorrect answers revealed many errors arise from inconsistencies or unclear information in the input data, which suggests that using full-text articles instead of titles and abstracts may improve results beyond the approach described here.

2.2.2. SLM Task B2: Untargeted compound occurrence data extraction. After assessing the extent to which language models can classify compound occurrences in a targeted manner, we next examined these systems' abilities with the same task in an untargeted way. For this process we used models that, as before, accept a system prompt with detailed instructions and a second prompt containing content with which to work. Our general approach was to provide a system prompt directing the model to read the input text (title/abstract) and write all experimentally supported compound occurrences in a Python dictionary format (e.g.: {'*Arabidopsis thaliana*': ['arabidiol', 'beta-sitosterol'], '*Brassica oleracea*': ['beta-sitosterol', 'alpha-amyrin']}). Thus, this task is considerably more complicated than the targeted approach. Due to this complexity, we conducted some preliminary tests to determine which of our 12 models might be suitable for this task. We found that the two large models and the two small ones were, respectively, too slow and too inaccurate to be feasible. For this reason, we proceeded with the six medium models as well as the two quantized 4-bit variants described in the previous section. Previous work has shown that the exact phrasing of system prompts can have substantial impacts on the accuracy of language model outputs (Razavi et al., 2025; Sclar et al., 2024), which included the context of phytochemical data processing (Knapp et al., 2025a). This phenomenon is the basis for prompt engineering. This untargeted task was inherently more complicated than the earlier described targeted approach, but further complications arose because we wanted a specific output format (the Python dictionary). We investigated a variety of prompts to determine how they might impact results from each model. As in the previous sections, to benchmark the ability of the models to perform this task, we again began by performing this task manually. We read 100 abstracts and wrote out the compound species associations reported in each in the JSON, or Python dictionary, format. This led to the identification of just over 400 compound occurrences across the 100 abstracts (Supplementary Materials S5). Next, we describe the performance of the 8 models and the 11 prompts on this untargeted compound occurrence extraction task with the 100 manually evaluated abstracts.

To begin, we carefully created a detailed system prompt and then employed a commercial large language model to produce 10 additional prompt variants that contained the same instructions but with different phrasings (all prompts included in Supplementary Materials S6). We then used each of the 11 prompts to instruct each of the eight models to write out all experimentally supported occurrences in each of the 100 manually evaluated abstracts. Next, we examined the ability of each model/prompt combination to provide results in a valid Python dictionary (the structure of the response needed to perform this data extraction task) and the speed at which each model/prompt combination could process the 100 abstracts. The percentage of responses from each model in answer to each prompt varied considerably, with some model/prompt combinations producing zero valid dictionaries and others generating 100% valid dictionaries (Figure 4a). Most model–prompt combinations produced >90% correctly structured responses, with some notable exceptions. Interestingly, qwen-2.5-14B-instruct struggled to consistently produce valid dictionary outputs, while its smaller sibling, qwen-2.5-7B-instruct, yielded over 90% valid dictionaries in most cases. This result breaks the trend of larger models being more proficient, as described in the previous section of this report. Phi-4 was the best model tested at this task since it returned 100% valid Python dictionaries, except for one response to prompt 8 (Figure 4a). We also observed variation among the prompts tested, with prompts 9, 10 and 4 eliciting higher proportions of valid responses across all the models than other prompts. We also examined the rate at which each model and prompt pairing could process queries. Rates ranged from about 200 references per hour to almost 1500 references per hour, with model size as the primary determinant of speed (Figure 4b). Different prompts sometimes caused variability in processing times for the same model, though these shifts were negligible compared to those driven by scale. Overall, the largest model, gemma-3-27B-instruct-unsloth, was the slowest. Meanwhile, phi-4-mini-instruct and qwen-2.5-7B-instruct performed the fastest, at rates around 1000 articles or references per hour. Altogether, the outcomes suggested that the phi-4 family models, along with qwen-2.5-7B-instruct combined with prompts 9, 10 and 4, were the most accurate for further detailed investigation. The four best performing models for producing valid Python dictionaries included the two fastest frameworks (phi-4-mini-instruct and qwen-2.5-7B-instruct), which showed that larger models do not always perform more proficiently than smaller versions.

In the previous section, we identified that the results from prompts 9, 10 and 4, in conjunction with phi-4, phi-4-mini-instruct and qwen-2.5-7B-instruct warranted further scrutiny. Therefore, we next examined the accuracy of occurrences generated by those models in response to those prompts. In contrast to our quantitative assessment of the models' ability to evaluate targeted compound instances, this broader approach allowed for quantifying only three response types: true positives (correct occurrences reported by a model), false positives (incorrect occurrences reported) and false negatives (correct occurrences missed by the model but found during manual evaluation, Figure 4c). Note that true negatives are not present in this untargeted analysis since the model is only asked to report existing occurrences, not to classify candidate occurrences. We quantified the number and category of each occurrence identified by each model in response to prompts 4, 9 and 10. We observed that using different system prompts led to only minor variations in the total correct versus incorrect instances flagged by a given model, but, interesting, that correct versus incorrect outputs varied greatly with respect to the number of species described in a given abstract (Figure 4d,e,h,i). Specifically, references involving more than four species appeared 'confusing' to the models, resulting in large numbers of inaccuracies from those sources (Figure 4d,h), while abstracts focused on one or two species typically yielded substantially more correct instances compared to incorrect ones (Figure 4d,h). Even so, the ratio of correct to incorrect responses typically generated from articles reporting on one or two species was roughly 2:1, an approximately 30% false positive rate.

To reduce the false positive rate observed during this untargeted compound occurrence extraction task, we introduced two types of filters. For the first filter, we programmatically compared the compound name reported in each occurrence against the PubChem database to check if it appeared among the entries. We removed all reported occurrences describing compounds missing from PubChem, which generally produced a bigger drop in incorrect results than in correct ones. The second filter relied on two language models identifying the same occurrence from a given abstract. Only those occurrences found by both, working independently, were kept, while partial matches (instances flagged by a single model but not recognized by another) were excluded. We tested this two-part filtering approach with two pairs of models: (i) one containing the most advanced model: phi-4 + qwen-2.5-7B-instruct and (ii) another featuring the two fastest options: phi-4-mini-instruct + qwen-2.5-7B-instruct. In both scenarios, the agreement filter yielded a marked decrease in inaccurate entries in the final dataset and only a small decline in valid ones (Figure 4d,h). Finally, to produce a dataset that reflects the lowest likely false positive rate for these models on the untargeted task at hand, we combined three filtering strategies: we restricted data to abstracts mentioning one or two species, retained only occurrences describing chemicals found in the PubChem database and kept only those occurrences that were independently detected from the same abstract by two different language models. Using this threefold approach, phi-4 + qwen-2.5-7B-instruct produced about 225 accurate occurrences and 25 inaccurate ones (an 11% error rate and ~55% yield, relative to the 400 occurrences found during manual inspection of the 100 abstracts, Figure 4f). Meanwhile, phi-4-mini-instruct + qwen-2.5-7B-instruct yielded 175 valid occurrences and around 20 erroneous findings (also an 11% error rate and ~44% yield/recall, Figure 4j). Thus, pairing two fastest models led to a dataset that was less comprehensive but maintained similar accuracy as a pair that contained a considerably larger and more sophisticated model.

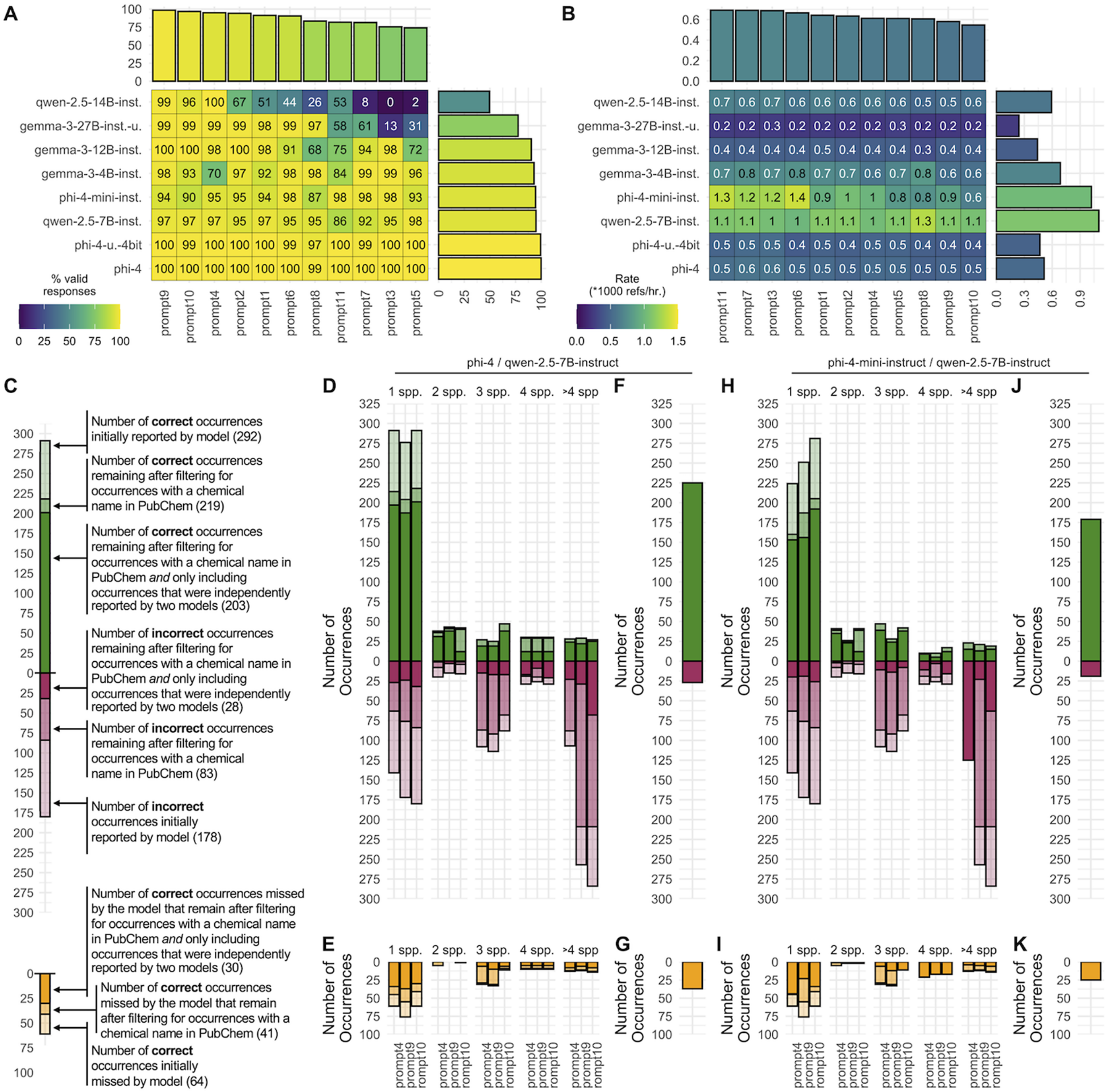

Figure 4. Performance of language models on a targeted compound occurrence data extraction task. (a) Heat map showing the per cent of outputs that contain valid python dictionaries (encoded with colour and written inside each box) from each language model (*y*-axis) in response to each prompt (*x*-axis). The marginal (i.e. top and right) plots show the mean per cent valid responses across all models for each prompt or across all prompts for each model. (b) Heat map showing the rate (in 1000 references per hour) of processing by each language model (*y*-axis) in response to each prompt (*x*-axis). The marginal (i.e. top and right) plots show the mean per cent valid responses across all models for each prompt or across all prompts for each model. (c) Guide describing how to interpret panels (d–k). Evaluation of occurrence data reported by language models (d/e/f/g: phi-4 and, in darkest bars, phi-4 in agreement with qwen-2.5-7B-instruct; h/i/j/k: phi-4-mini-instruct and, in darkest bars, phi-4-mini-instruct in agreement with qwen-2.5-7B-instruct). (d) and (h) show the number of correct occurrences (true positives, positive *y*-axis) and incorrect occurrences (false positives, negative *y*-axis) reported, as indicated in panel c. (e) and (i) show the number of correct occurrences (false negatives, negative *y*-axis) reported, as indicated in panel c. (f) and (j) show the number of correct occurrences (true positives, positive *y*-axis) and incorrect occurrences (false positives, negative *y*-axis) reported after filtering for occurrences whose compounds are in PubChem and were agreed upon by the two models. (g) and (k) show the number of correct occurrences missed by the models after PubChem and agreement filtering (false negatives, negative *y*-axis). In (d–k), bar orientation emphasizes desired model behaviour: bars pointing upwards indicate correct model responses (desired behaviour), while bars pointing down indicate incorrect model responses or correct answers not reported by the model (undesired behaviour).

3. Conclusions and future directions

Here, we evaluated the ability of small language models to perform two major tasks: to numerically score references based on their relevance to a given topic (SLM Task A) and to extract structured data from unstructured inputs in both a targeted (SLM Task B1) and untargeted fashion (SLM Task B2). Our efforts showed that a small language model could rapidly and effectively score references in a set so that a threshold score could be used as a filter to substantially enrich that set for articles of interest (Section 2.1). Limitations arose when handling edge cases, with highly tailored, task-specific prompts emerging as a possible approach to address those shortcomings. When using small language models to classify candidate compound occurrences as true or false, we observed that if an abstract directly reports the detection of a particular compound in a specific plant species, the models nearly always label the candidate occurrence correctly (Section 2.2.1). In this task, however, a trade-off did appear between accuracy and model size (parameter count) and compute requirements. Among the mistakes noted (false positives as low as 5% and false negatives as low as 2% for certain models), these misclassifications were as often tied to convoluted and unclear writing in the input abstract as they were to outright model errors. For extracting compound occurrence information from unstructured text in an untargeted manner, we found small language models to be effective, though choosing a suitable model and pipeline strategy proved more challenging than earlier tasks (Section 2.2.2). We discovered that prompt engineering, selecting a model and filtering reported detections by cross-referencing chemical databases, along with requiring two small language models to independently agree on an occurrence, yielded the best reporting statistics (~10% false positives and ~50% yield). Of note, this relatively low yield arises because many correct associations are filtered out, essentially sacrificed, to lower the false positive rate. Regarding all tasks considered, more advanced prompting techniques (e.g. chain-of-thought prompting (Wei et al., 2022) or model distillation (Hinton et al., 2015; Sanh et al., 2019)) could reduce error rates further and improve yield/recall. In addition, future model releases, including small reasoning models, may also address these limitations. Finally, we will note that many abstracts we worked with here presented problems for humans and language models alike by failing to contain clear and concise information. We read hundreds of abstracts for the present project. Fully understanding many abstracts in a timely fashion was extremely difficult due to long, convoluted sentences, the presentation of connected data types (e.g. plants and compounds) in multiple sentences spread throughout a long abstract, the use of compound numbers or abbreviations instead of compound names, poor grammar and so forth. In a variety of cases, we were surprised that the language models performed reasonably well, while humans needed considerable time to understand the same abstracts.

Overall, though the approaches here represent a considerable advance over manual curation (at least, with respect to the creation of large databases, where speed is a prime consideration), a substantial amount of plant chemical occurrence data will still not be retrieved from the literature using the techniques presented here. One important step forward will be the development of pipelines that can handle articles reporting occurrence data from dozens of species, including in tabular format. In addition, further attempts towards occurrence databases and in fact scientific endeavours in general, need literature databases that include the full-text files along with reference citations and abstracts. The separation of the full text from the citations seems to be a systematic and legal barrier that needs to be overcome. The expanded posting of preprints is suggested as a potential, albeit partial, solution to this issue. In addition to the tasks we quantitatively evaluated here, we also experimented with several versions of Microsoft's Phi-4 model to conduct multiple activities related to reference citations (e.g. species name extraction, compound number or plant number extraction, etc.) and found that the models could perform a range of additional functions, suggesting versatility and application in order domains. In our case, these functions have allowed us to identify publications that most likely contain extensive tabular data in the full text, flagging them for analysis by a pipeline suitable for such reports. Finally, in our efforts, we found that filtering capabilities such as those provided by SciFinder® and EndNote^{TM} showed usefulness in a somewhat orthogonal way to the value of the small language model scores. For example, in our case, we were able to eliminate many articles of low relevance to our case studies using EndNote™ keyword filters. As these commercial software tools and other related programmes are outfitted with language model ('artificial intelligence') capabilities, it will be important to evaluate and incorporate those features into discipline-specific workflows. We strongly encourage the scientific community to look for new versions of their favourite research tools that incorporate language model features and to experiment and empirically test and report on such functionality in field-specific tasks as they emerge.

4. Methods

Literature searches were conducted with CAS SciFinder®. SciFinder® searches were conducted by entering the compound CAS Registry® number from the SUBSTANCE menu and then working with all the references that were assigned to this Registry number. SciFinder® references were downloaded as 'tagged' text files. The 'tagged' text file selection provides numerous fields including the CAS Registry numbers for all compounds discussed in a given article. Multiple tagged files were downloaded for each compound (according to year ranges) since the SciFinder® software limits an individual tagged export file to 100 citations. SciFinder limits the number of citations that can be exported in one file to 100. Thus, for a compound such as alpha-amyrin with 4344 SciFinder references, the downloading of all references was not possible. If the number of filtered references was greater than 400, the word 'plant' was entered into the 'search within results'. Thus, only English-language journal references that corresponded to the 'search within results' term 'plant' were downloaded (1744 references, in the example of alpha-amyrin). PubMed® searches for the six triterpenoids were also conducted based on their major common names (not all synonyms were used). These PubMed® searches were conducted with the compound names shown at the top of Figure 1 since PubMed® does not generally recognize CAS Registry® numbers. PubMed® files were downloaded as PubMed (NLM) files. Of note is that PubMed® provides automated access to its search and abstract download services through a REST API and various language-specific packages like R and Trez. These tools could be leveraged in the future to further streamline literature analysis projects and automate data extraction and tabulation.

EndNoteTM Version 21.5 (https://endnote.com/) was used to import and combine the sets of 'tagged' SciFinder® export text files for each compound into an individual EndNoteTM compound folders (with the 'discard duplicate' feature turned on). Furthermore, EndNoteTM 'Smart Groups' were set up for each of the six triterpenoids, which included the CAS Registry® number and multiple names for each compound (i.e. synonyms). The references in each of the six Smart Groups were then added to the corresponding original six triterpenoid folders (with automatic elimination of duplicates). As noted earlier, some plants contained more than one of the six triterpenoids. These EndNoteTM operations ensured that any references that might have been missed in a given SciFinder® compound search, but included in another compound search, would end up in the appropriate folders (i.e. one reference might be in more than one compound folder). In EndNote®, the user can select scores of references and then right-click on 'Find full text'. EndNote will then automatically download the PDF files for each reference that cites a journal for which the user's institution has a subscription or an open-source journal. However, in some cases, software blocks (e.g. the 'Are you a human filter?') prevent the downloading of some files. In our case at our institution, EndNote downloads approximately 40–50% of the PDF files for the selected references.

All manual evaluation of reference relevance ('reports an occurrence', 'maybe reports an occurrence', 'does not report an occurrence'), manual evolution of candidate occurrences (targeted) and manual extraction of associations (untargeted) was performed by opening the list of references in Microsoft Excel and entering the manual annotations into a new column. References were labelled as 'maybe reports an occurrence' if they mentioned specific plant species and the isolation of multiple compounds from the species but did not mention the specific compounds' names in the abstract. While the likelihood of a plant/compound association appearing in the full article was high, we nevertheless conservatively chose to label these types of citations as 'maybe reports an occurrence' until the full-text article file could be evaluated. An 'maybe' example is:

'Medicinal attributes of *Solanum capsicoides* All.: an antioxidant perspective. *Int. J. Pharm. Sci. Res.* 12(5): 2810–2817. The study evaluates the medicinal efficacy of *Solanum capsicoides* fruits as an antioxidant. Fruit extracts were prepared using acetone, ethanol, HCl and water [. . .] A neg. correlation was observed between the pigments, anthocyanins and carotenoids, with DPPH and CUPRAC activity. [. . .] From this study, it can be considered that the phenolics present in the fruits contribute to the characteristic antioxidant property'.

The Facebook BART-Large-MNLI zero-shot classification model (https://huggingface.co/facebook/bart-large-mnli) was applied to the individual sets of compound reference citations in the EndNoteTM database. The model was run on a single NVIDIA GV100GL [Quadro GV100] GPU. First, the set of references in the curated EndNoteTM folder for a given compound was selected and exported from this folder to a text file (with the 'annotated' style selected). This text file was then imported into an Excel file (e.g. with the legacy 'get text from file' Excel wizard. The resulting Excel sheet was then modified so that each reference citation (author/year/journal/abstract) was contained in one cell and all cells resided in one column. This Excel sheet, which contained all the reference citations for a given compound, was then saved as a CSV UTF-8 (Comma delimited) file. This CSV file was used via JupyterLab (https://jupyter.org/, operating in a WINDOWS 11 environment) and a custom Python program (full code in Supplementary Materials S8). System prompt-accepting chat language models were downloaded from HuggingFace.co and run on a single NVIDIA GV100GL [Quadro GV100] GPU using custom code (full code provided in Supplementary Materials S8). Calculation of precision, recall and F1 scores as well as plotting were performed in R. Additional system prompts for the prompt engineering reported in Section 2.2.2 were generated by OpenAI's 04-mini-high language model using the ChatGPT browser interface.

Open peer review. To view the open peer review materials for this article, please visit http://doi.org/10.1017/qpb.2025.10021.

Acknowledgements

The authors acknowledge the support of the University of Minnesota Duluth Chemistry and Biochemistry Department. During the writing of this manuscript, the large language model o4-mini-high from OpenAI was utilized for copy editing to suggest alternative sentence structures and word choices that, in our view, enhanced grammar and readability. Finally, we collectively acknowledge that the University of Minnesota Duluth is located on the traditional, ancestral and contemporary lands of Indigenous people. The University resides on land that was cared for and called home by the Ojibwe people, before them the Dakota and Northern Cheyenne people and other Native peoples from time immemorial. Ceded by the Ojibwe in an 1854 treaty, this land holds great historical, spiritual and personal significance for its original stewards, the Native nations and peoples of this region. We recognize and continually support and advocate for the sovereignty of the Native nations in this territory and beyond. By offering this land acknowledgement, we affirm tribal sovereignty and will work to hold the University of Minnesota Duluth accountable to American Indian peoples and nations.

Competing interest. The authors declare none.

Data availability statement. All data and code used in this study are available for free as a supplement to this document.

Author contributions. LB and ARO conceived and designed the study, gathered data and wrote the manuscript.

Funding statement. This work was funded by startup funds granted to Lucas Busta from the University of Minnesota Duluth Swenson College of Science and Engineering.

Supplementary materials. The supplementary material for this article can be found at http://doi.org/10.1017/qpb.2025.10021.

References

Abdin, M., Eldan, R., Javaheripi, M., Li, Y., Price, E., Shah, S., Yu, D., Aneja, J., Gunasekar, S., Kauffmann, P., Liu, W., Gd, R., Wang, X., Zhang, C., Behl, H., Harrison, M., Lee, J. R., Mendes, C. C. T., Saarikivi, O., . . . **Wu, Y.** (2024). Phi-4 technical report. arXiv:241208905v1 [csCL], 12 Dec 2024.

Agathokleous, E., Rillig, M. C., Penuelas, J., & **Yu, Z.** (2024). One hundred important questions facing plant science derived using a large language model. *Trends in Plant Science*, **29**(2), 210–218. https://doi.org/10.1016/j.tplants.2023.06.008.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . **Amodei, D.** (2020). Language models are few-shot learners. arXiv.

Busta, L., Hall, D., Johnson, B., Schaut, M., Hanson, C. M., Gupta, A., Gundrum, M., Wang, Y.,. A., & **Maeda, H.** (2024). Mapping of specialized metabolite terms onto a plant phylogeny using text mining and large language models. *The Plant Journal*, **2024-7-8**. https://doi.org/10.1111/tpj.16906.

Chen, Y., de Bruyn Kops, C., & **Kirchmair, J.** (2017). Data resources for the computer-guided discovery of bioactive natural products. *Journal of Chemical Information and Modeling*, **57**(9), 2099–2111. https://doi.org/10.1021/acs.jcim.7b00341.

Dalal, A., Ranjan, S., Bopaiah, Y., Chembachere, D., Steiger, N., Burns, C., & **Daswani, V.** (2024). Text summarization for pharmaceutical sciences using hierarchical clustering with a weighted evaluation methodology. *Scientific Reports*, **14**(1), 20149. https://doi.org/10.1038/s41598-024-70618-w.

Gallo, K., **Kemmler, E.**, **Goede, A.**, **Becker, F.**, **Dunkel, M.**, **Preissner, R.**, & **Banerjee, P.** (2023). SuperNatural 3.0-a database of natural products and natural product-based derivatives. *Nucleic Acids Research*, **51**(D1), D654–D659. https://doi.org/10.1093/nar/gkac1008.

Gemma, T., **Kamath, A.**, **Ferret, J.**, **Pathak, S.**, **Vieillard, N.**, **Merhej, R.**, **Perrin, S.**, **Matejovicova, T.**, **Ramé, A.**, **Rivière, M.**, **Rouillard, L.**, **Mesnard, T.**, **Cideron, G.**, **J-b, G.**, **Ramos, S.**, **Yvinec, E.**, **Casbon, M.**, **Pot, E.**, **Penchev, I.**, . . . **Hussenot, L.** (2025). Gemma 3 technical report. arXiv.

Guo, Z., **Wnag, Y.**, **Wang, P.**, & **Yu, P.** (2023). Improving small language models on PubMedQA via generative data augmentation. arXiv:230507804v4, [csCL] 1 Aug 2023.

HintonG, **VinyalsO**, **DeanJ** (2015) Distilling the knowledge in a neural network. 1503.02531.

Jin, Q., **Leaman, R.**, & **Lu, Z.** (2024). PubMed and beyond: Biomedical literature search in the age of artificial intelligence. *eBioMedicine*, **100**, 104988. https://doi.org/10.1016/j.ebiom.2024.104988.

Kaplan, J., **McCandlish, S.**, **Henighan, T.**, **Brown, T. B.**, **Chess, B.**, **Child, R.**, **Gray, S.**, **Radford, A.**, **Wu, J.**, & **Amodei, D.** (2020). Scaling laws for neural language models. *arXiv*.

Knapp, R., **Johnson, B.**, & **Busta, L.** (2025). Advancing plant metabolic research by using large language models to expand databases and extract labelled data. *Applications in Plant Sciences*, p.e70007. https://doi.org/10.1101/2024.11.05.622126.

Lam, H. Y. I., **Ong, X. E.**, & **Mutwil, M.** (2024). Large language models in plant biology. *Trends in Plant Science*, **29**(10), 1145–1155. https://doi.org/10.1016/j.tplants.2024.04.013.

LepagnolP, **GeraldT**, **GhannayS**, **SeranC**, **RossetS** (2024) Small language models are good too: An empirical study of zero-shot classification. arXiv:240411122v1 [csAI] 17 Apr 2024. https://arxiv.org/abs/2404.11122?form=MG0AV3

Lewis, M., **Liu, Y.**, **Goyal, N.**, **Ghazvininejad, M.**, **Mohamed, A.**, **Levy, O.**, **Stoyanov, S.**, & **Zettlemoyer, L.** (2019). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv:191013461v1 [csCL], 29 Oct 2019.

Maronikolakis, A., **Dufter, P.**, & **Schütze, H.** (2021). BERT cannot align characters. arXiv preprint arXiv:2109.09700.

Nguyen-Vo, T.-H., **Nguyen, L.**, **Do, N.**, **Nguyen, T.-N.**, **Trinh, K.**, & **Cao, H. L.** (2020). Plant metabolite databases: From herbal medicines to modern drug discovery. *Journal of Chemical Information and Modeling*, **60**(3), 1101–1110. https://doi.org/10.1021/acs.jcim.9b00826

Qwen: An, Y., **Baosong, Y.**, **Beichen, Z.**, **Binyuan, H.**, **Bo, Z.**, **Bowen, Y.**, **Chengyuan, L.**, **Dayiheng, L.**, **Fei, H.**, **Haoran, W.**, **Huan, L.**, **Jian, Y.**, **Jianhong, T.**, **Jianwei, Z.**, **Jianxin, Y.**, **Jiaxi, Y.**, **Jingren, Z.**, **Junyang, L.**, **Kai, D.**, . . . **Zihan, Q.** (2025). Qwen2.5 technical report :contentReference[oaicite:1]{index=1}. https://doi.org/10.48550/arXiv.2412.15115.

Razavi, A., **Soltangheis, M.**, **Arabzadeh, N.**, **Salamat, S.**, **Zihayat, M.**, & **Bagheri, E.** (2025). Benchmarking prompt sensitivity in large language models. https://arxiv.org/abs/2502.06065.

Riordan, B. B., **Albert, A.**, **He, Z.**, **Nibali, A.**, **Anderson-Luxford, D.**, & **Kuntsche, E.** (2024). How to apply zero-shot learning to text data in substance use research: An overview and tutorial with media data. *Addiction*, **119**(5), 951–959. https://doi.org/10.1111/add.16427

Rutz, A., **Sorokina, M.**, **Galgonek, J.**, **Mietchen, D.**, **Willighagen, E.**, **Gaudry, A.**, **Graham, J. G.**, **Stephan, R.**, **Page, R.**, **Vondrášek, J.**, **Steinbeck, C.**, **Pauli, G. F.**, **Wolfender, J.-L.**, **Bisson, J.**, & **Allard, P.-M.** (2022). The LOTUS initiative for open knowledge management in natural products research. *eLife*, **11**. https://doi.org/10.7554/eLife.70780.

Sanh, V., **Debut, L.**, **Chaumond, J.**, & **Wolf, T.** (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108.

Sarumi, O. A., & **Heider, D.** (2024). Large language models and their applications in bioinformatics. *Computational and Structural Biotechnology Journal*, **23**, 3498–3505. https://doi.org/10.1016/j.csbj.2024.09.031.

Sclar, M., **Choi, Y.**, **Tsvetkov, Y.**, & **Suhr, A.** (2024). Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. arXiv.

Shiu, S.-H., & **Lehti-Shiu, M. D.** (2024). Assessing the evolution of research topics in a biological field using plant science as an example. *PLoS Biology*, **22**(5), e3002612. https://doi.org/10.1371/journal.pbio.3002612.

Simon, E., **Swanson, K.**, & **Zou, J.** (2024). Language models for biological research: A primer. *Nature Methods*, **21**(8), 1422–1429. https://doi.org/10.1038/s41592-024-02354-y.

Sorokina, M., & **Steinbeck, C.** (2020). Review on natural products databases: Where to find data in 2020. *Journal of Cheminformatics*, **12**(1), 20. https://doi.org/10.1186/s13321-020-00424-9.

Stevenson, C. E., **Pafford, A.**, **van der Maas, H. L.**, & **Mitchell, M.** (2024). Can large language models generalize analogy solving like people can? arXiv preprint arXiv:2411.02348.

Tay, D. W. P., **Yeo, N. Z. X.**, **Adaikkappan, K.**, **Lim, Y. H.**, & **Ang, S. J.** (2023). 67 million natural product-like compound database generated via molecular language processing. *Scientific Data*, **10**(1), 296. https://doi.org/10.1038/s41597-023-02207-x.

Wei, J., **Wang, X.**, **Schuurmans, D.**, **Bosma, M.**, **Xia, F.**, **Chi, E.**, **Le, Q. V.**, & **Zhou, D.** (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, **35**, 24824–24837.

Xie, T., **Song, S.**, **Li, S.**, **Ouyang, L.**, **Xia, L.**, & **Huang, J.** (2015). Review of natural product databases. *Cell Proliferation*, **48**(4), 398–404. https://doi.org/10.1111/cpr.12190.

Yang, B., **Mao, J.**, **Gao, B.**, & **Lu, X.** (2019). Computer-assisted drug virtual screening based on the natural product databases. *Current Pharmaceutical Biotechnology*, **20**(4), 293–301. https://doi.org/10.2174/1389201020666190328115411.

Yin, W., **Hay, J.**, **Rother, D.** (2019) Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 3914–3923). Hong Kong, China, November 3–7, 2019

Zeng, T., **Li, J.**, & **Wu, R.** (2024). Natural product databases for drug discovery: Features and applications. *Pharmaceutical Science Advances*, **2**. https://doi.org/10.1016/j.pscia.2024.100050.

Zhu, X., **Li, J.**, **Liu, Y.**, **Ma, C.**, & **Wang, W.** (2024). Distilling mathematical reasoning capabilities into small language models. *Neural Networks: The Official Journal of the International Neural Network Society*, **179**, 106594. https://doi.org/10.1016/j.neunet.2024.106594.