

AI-Driven Multimodal Validation of Chemical Formulas. Insights from Gemini Thinking Log

Evgeny Markhasin
Lobachevsky State University of Nizhny Novgorod
<https://orcid.org/0000-0002-7419-3605>
<https://linkedin.com/in/evgenymarkhasin>

Abstract

This work explores prompt engineering strategies for the validation of chemical formulas in scholarly manuscripts using Large Language Models (LLMs). Building upon previous work on AI-driven peer review, we investigate the efficacy of various prompting techniques, from basic queries to complex framework, for identifying errors in chemical identifiers. Using a challenging test paper known to contain subtle errors, we observe that simple, direct prompts yield unreliable results. More structured prompts show incremental improvements but still lack robustness. Key insights are derived from analyzing Gemini “thinking logs”; this analysis indicates potential challenges related to the models’ inherent error-correction tendencies that can mask errors in input text. We then adapt previously demonstrated advanced approach employing LLM context conditioning, combined with explicit instructions for multimodal analysis of figures yielding a proof-of-concept prompt for formula validation. This method demonstrates improved reliability in identifying errors both in text and figures. The findings highlight the potential of sophisticated prompting and multimodal capabilities for chemical entity validation. This work is a supplementary material for [arXiv:2505.03332 \[cs.AI\]](https://arxiv.org/abs/2505.03332) (DOI: [10.48550/arXiv.2505.03332](https://doi.org/10.48550/arXiv.2505.03332)).

1. Introduction

Previous work explored the application of advanced prompt engineering techniques, specifically Persistent Workflow Prompting (PWP), for the critical analysis of scholarly manuscripts in experimental chemistry, using a specific test paper [test paper](#) as a challenging case study. This article extends that investigation by focusing on a common yet critical task: the validation of chemical formulas within scientific texts. The same test paper, a 44-page document including its Supporting Information (SI), serves as a pertinent and challenging test case for this task due to known, subtle errors. For instance, page S-8 of the test paper’s SI presents the formula for *ferrous ammonium sulfate* as $Fe(NH_4)_2SO_4$, which incorrectly omits one sulfate group (the correct formula for ferrous ammonium sulfate, Mohr’s salt, is $(NH_4)_2Fe(SO_4)_2 \cdot 6H_2O$ or $Fe(NH_4)_2(SO_4)_2$ - anhydrous). This exploration details the iterative process of developing prompts to reliably identify such a chemical “needles in a haystack”.

2. Approaches to Chemical Formula Validation with LLMs

2.1. Direct Prompting Attempts and Initial Observations

Initial efforts involved testing basic prompts with leading LLMs (Gemini Advanced 2.5 Pro and ChatGPT Plus o3). A simple prompt such as: Find mistakes in chemical formulas and names. produced highly inconsistent and generally unreliable results, occasionally identifying the target error but lacking systematic accuracy.

A more structured prompt was subsequently developed:

Chemical Formula Extraction and Validation from PDF

Execute the following task step-by-step:

1. Extract each and every chemical formula from the attached PDF.
2. For each extracted formula, extract every directly associated chemical name included in the text, if any.
3. For each extracted formula and associated names, consider if the chemical formula and associated names are correct and flag every formula/names combination that contains any errors.
4. For each flagged item, read the source PDF again and confirm that the item was extracted exactly. In case of any extraction errors, analyze the corrected item and consider if the flag should be removed.
5. Create a Markdown table that should include every flagged formula/names, clear description of any problems, corrected version, and clear reference location of the flagged items.

This refined prompt improved the quality of the output and increased the frequency of identifying the target error in $Fe(NH_4)_2SO_4$ [GeminiNaiveAnalysis](#). However, reliability remained a significant issue.

An interesting observation arose from a slightly modified version of this prompt [GeminiNaiveAnalysisReaction](#). The LLM not only extracted chemical formulas but also identified and flagged an imbalanced chemical reaction scheme from page S-8 of the SI, immediately following the problematic ferrous ammonium sulfate formula. The reaction of Fe(II) and hydrogen peroxide was flagged due to an imbalance in iron atoms (3 Fe on the left, 2 Fe on the right). This error, previously overlooked by the author during manual review, was correctly identified and a balanced version proposed by the LLM. This highlighted the LLM's potential to uncover errors beyond the specific target, even when the prompt was focused on "formulas".

2.2. Challenges with LLM Default Behaviors

Examination of Gemini's "Show thinking" logs (a feature providing insight into the model's processing steps) for various runs revealed a consistent pattern. The LLM often correctly extracted the target formula (e.g., " $Fe(NH_4)_2SO_4$: Ferrous ammonium sulfate (Mohr's salt)"). However, in the subsequent validation step, it would sometimes erroneously mark the pair as correct, e.g.:

Identified Chemical Formulas/Names and Initial Validation:

...(Other log lines)...

$Fe(NH_4)_2SO_4$: Ferrous ammonium sulfate. Correct.

This behavior likely stems from a core strength of LLMs: their inherent capability for error correction and understanding intent despite minor inaccuracies in the input. For instance, querying `What is the capital of grate britain?` typically yields "London", with the misspelling of "Great Britain" being automatically corrected. While usually beneficial, this default error tolerance becomes a hindrance when the objective is to *detect* such errors. This phenomenon is analogous to the "input bias" discussed in the PWP preprint, where the LLM's tendency to accept input information as factual needs to be actively countered for critical evaluation.

In the context of formula validation, the LLM's natural inclination to reconcile a slightly incorrect formula with its correct accompanying name complicates direct "contrasting" methods (as per Point 3 in the structured prompt above). To address this problem, strategies must either suppress this error-correction tendency in a controlled manner or employ more complex error detection algorithms that avoid direct comparison of such "complementary" yet potentially mismatched entities.

An early attempt at the latter approach involved a multi-step generation and comparison process (see “Appendix. Chemical Formula Analysis - Generated Formulas and Names” for the prompt and an example of LLM response). The intent was for the LLM to:

```
Extract a chemical formula from text -> Generate a chemical name -> Generate a chemical formula -> Compare the extracted formula with the newly generated formula.
```

This approach, while, possibly, better, still proved unreliable in practice, although some runs did yield interesting outputs, including references to external chemical databases.

3. Advanced Validation using Context Conditioning and Multimodal Analysis

Given the limitations of direct and simple generative approaches, a more robust strategy was adopted, leveraging the context conditioning principles outlined in the PWP preprint. The *PeerReviewPrompt* detailed in that work successfully mitigated input bias through comprehensive context setting.

A new prompt, the [ChemicalFormulasValidationPrompt](#), was developed by adapting core sections from the *PeerReviewPrompt*:

- Sections **I-III** (Core Objective, Persona, Context: Framework for Critical Review) were largely retained to establish the analytical mindset and operational guidelines.
- Section **V** (Final Instructions for Interaction) was kept to ensure consistent LLM behavior.
- Section **IV.A** (Foundational Principles & Workflow Application) was adapted.
- A new **Chemical Identifier Analysis** subsection was introduced into Section **IV** specifically for formula and name validation, including explicit instructions for multimodal analysis of figures.

This PWP-style prompt can be submitted to the LLM initially (with or without the target PDF). A subsequent query, such as:

```
Perform comprehensive Chemical Identifier Analysis
```

with the target PDF attached, then initiates the validation workflow.

Demonstration AI chats using this approach with Gemini 2.5 Pro (via Google AI Studio, which often shows enhanced performance over the standard subscription-based Gemini app interface) [GeminiAnalysis](#) and ChatGPT Plus o3 [ChatGPTAnalysis](#) showed improved robustness and reliability. The current [ChemicalFormulasValidationPrompt](#) instructs the LLM to output a table of *all* extracted chemical formulas, detailing any identified issues and providing corrected versions, or a check mark if no error is found.

Crucially, this advanced approach consistently identified the target error in $Fe(NH_4)_2SO_4$. Furthermore, due to the explicit instruction to perform multimodal analysis (specifically, analyzing figures), the Gemini model also reproducibly identified another error that had been missed in prior manual reviews:

In Figure 2c (page 235 of the main text of the [test paper](#)), an NMR spectrum is presented with a label $(CH_3)_3Si_2O$ for hexamethyldisiloxane. The correct formula is $((CH_3)_3Si)_2O$ or $(CH_3)_6Si_2O$. Gemini correctly flagged the inconsistency in the depicted formula.

While occasional false positives were observed in the outputs from both models, the PWP-based prompt with multimodal instructions demonstrated a marked improvement in reliably detecting subtle chemical formula errors, including those embedded within figures. While ChatGPT o3 model is advertised as being capable of multimodal analysis, it failed to identify the error in the figure.

4. Conclusions

Validating chemical formulas within scientific texts using LLMs presents distinct challenges, as illustrated by the unreliable performance of simple or directly structured prompting strategies and insights from LLM analysis logs. These logs, for instance, indicate potential challenges related to the models' inherent error-correction tendencies, which can mask the very errors targeted for detection and contribute to the observed unreliability.

Preliminary tests demonstrated the efficacy of advanced LLM context conditioning in overcoming this challenge. Specifically, the presented *ChemicalFormulasValidationPrompt*, adapted from the previously developed PWP-based *PeerReviewPrompt*, reproducibly identified the target chemical formula error using two frontier, generally available LLM models. Furthermore, reproducible multimodal analysis was demonstrated with the Gemini 2.5 Pro model, where a defective chemical formula within a figure was successfully identified by relating it to its name extracted from the figure caption, and subsequently corrected. Conversely, the same tests failed to elicit effective multimodal analysis for this specific task with the ChatGPT Plus o3 model.

The demonstrated robustness of context conditioning in managing LLM behaviors - particularly those contributing to input bias and error suppression, as evidenced in this study and the previous PWP preprint - suggests potential for broader applications. For instance, similar PWP-informed techniques could prove useful in fields such as medical AI, especially for workflows involving the critical processing and validation of information from individual patient records. Another promising application lies in the extraction and validation of information from semi-structured, unstructured, or poorly structured sources, such as those found in pharmaceutical or technical documentation, where the ability to suppress silent error correction by LLMs may be of paramount importance.

It is important to acknowledge, however, that the present study itself is based on a limited testing scope. Consequently, while the presented prompt shows promise as a proof-of-concept, future work should focus on more extensive testing across diverse datasets and further refinement of these PWP-informed methods to enhance accuracy and ensure wider applicability in complex scientific content analysis and validation.

5. References

- **PWP:** Markhasin, E. (2025). *AI-Driven Scholarly Peer Review via Persistent Workflow Prompting, Meta-Prompting, and Meta-Reasoning*. arXiv preprint. <https://arxiv.org/abs/2505.03332>.
 - **Test paper:** Prasad, B., Lewis, A. R., & Plettner, E. (2011). Enrichment of H217O from Tap Water, Characterization of the Enriched Water, and Properties of Several 17O-Labeled Compounds. *Analytical Chemistry*, 83(1), 231–239. <https://doi.org/10.1021/ac1022887>.
 - **ChemicalFormulasValidationPrompt:** Chemical Formulas Validation Prompt. https://osf.io/nq68y/files/osfstorage?view_only=fe29ffe96a8340329f3ebd660faedd43.
 - **GeminiNaiveAnalysis:** Shared Gemini Chat Log. <https://g.co/gemini/share/a62c19799936>.
 - **GeminiNaiveAnalysisReaction:** Shared Gemini Chat Log. <https://g.co/gemini/share/70cd2d7b296b>.
 - **GeminiAnalysis:** Shared Google AI Studio Chat Log. <https://aistudio.google.com/app/prompts/1CUdXeLstQQp1y0UZ9R9ZvaBuhy6CNW1>.
 - **ChatGPTAnalysis:** Shared ChatGPT Chat Log. <https://chatgpt.com/share/681f5b9f-375c-8004-b2f4-294c75371945>.
-

Appendix. Chemical Formula Analysis - Generated Formulas and Names

Prompt

Execute the following task step-by-step:

1. Extract chemical formulas of each and every chemical species containing at least two elements EXACTLY as they appear in the attached PDF.
2. For each extracted chemical formula generate associated name.
3. Convert each generated name to generated chemical formula.
4. For each generated chemical formula, determine if it matches previously extracted formula.

Response Example (Illustrative)

Chemical Species Analysis from PDF

Here's an analysis of the chemical species found in the provided PDF document, "Enrichment of H2 17O from Tap Water, Characterization of the Enriched Water, and Properties of Several 17O-Labeled Compounds."

Extracted Chemical Formulas, Generated Names, and Formula Comparison

The following table details the chemical formulas extracted directly from the PDF (containing at least two elements), their generated chemical names, the chemical formulas regenerated from these names, and a comparison to determine if the original and regenerated formulas match.

#	Extracted Formula (from PDF)	Generated Chemical Name	Regenerated Chemical Formula (from Name)	Matches Original?	Source(s) for Name/Regenerated Formula
1	H217O	Water-17O	H217O	Yes	Sigma-Aldrich , Sigma-Aldrich
2	CDCl3	Deuterated chloroform	CDCl3	Yes	EMBL-EBI , Wikipedia
3	Na17OH	Sodium hydroxide-17O	Na17OH (or HNa17O)	Yes	Name manually determined due to ambiguous search; Formula from ChemSpider
4	H217O2	Hydrogen peroxide-17O2	H217O2	Yes	Sigma-Aldrich , ChemSpider
5	H2O2	Hydrogen peroxide	H2O2	Yes	Britannica , US EPA , AccessScience
6	NO	Nitric oxide (Nitrogen monoxide)	NO	Yes	EMBL-EBI
7	H2O	Water (Oxidane)	H2O	Yes	Wikipedia , PubChem
8	H218O	Water-18O	H218O	Yes	PubChem , NIST WebBook
9	KH2PO4	Potassium dihydrogen phosphate	KH2PO4	Yes	American Chemical Society , EMBL-EBI

10	K ₂ HPO ₄	Dipotassium phosphate	K ₂ HPO ₄	Yes	DrugBank , Wikipedia , PubChem
11	KCl	Potassium chloride	KCl	Yes	BYJU'S , Wikipedia
12	RuO ₂	Ruthenium(IV) oxide	RuO ₂	Yes	Wikipedia
13	NaI	Sodium iodide	NaI	Yes	Wikipedia
14	H ₂ 16O	Water-16O	H ₂ 16O	Yes	Sigma-Aldrich , SSHADE
15	CH ₄	Methane	CH ₄	Yes	EMBL-EBI , Wikipedia
16	HCl	Hydrogen chloride	HCl	Yes	Wikipedia , EMBL-EBI
17	NaOH	Sodium hydroxide	NaOH	Yes	Wikipedia , The Chlorine Institute
18	NaCl	Sodium chloride	NaCl	Yes	EMBL-EBI , Taylor & Francis
19	Na ₂ CO ₃	Sodium carbonate	Na ₂ CO ₃	Yes	Home Science Tools , Wikipedia
20	NaHCO ₃	Sodium bicarbonate	NaHCO ₃	Yes	Wikipedia , EBSCO Research Starters
21	Na ₂ SO ₄	Sodium sulfate	Na ₂ SO ₄	Yes	PubChem , Wikipedia
22	MgSO ₄	Magnesium sulfate	MgSO ₄	Yes	FAO
23	CF ₃	Trifluoromethyl radical	CF ₃	Yes	PubChem
24	NH ₂	Amino radical	NH ₂ (or NH ₂ ·)	Yes	Wikipedia , Wikipedia
25	CD ₃ CN	Acetonitrile-d ₃	CD ₃ CN	Yes	Merck Millipore , Sigma-Aldrich
26	CCl ₄	Carbon tetrachloride	CCl ₄	Yes	PubChem
27	KMnO ₄	Potassium permanganate	KMnO ₄	Yes	PubChem , DermNet
28	Fe(NH ₄) ₂ SO ₄	Ferrous ammonium sulfate	(NH ₄) ₂ Fe(SO ₄) ₂ or Fe(NH ₄) ₂ (SO ₄) ₂ ·6H ₂ O	No (Original likely a typo or refers to a different species than common Ferrous Ammonium Sulfate / Mohr's Salt which is a double salt)	EMBL-EBI , CAMEO Chemicals - NOAA
29	KSCN	Potassium thiocyanate	KSCN	Yes	Wikipedia , Spectrum Chemical

30	Fe(SCN)3	Iron(III) thiocyanate	Fe(SCN)3 (or C3FeN3S3)	Yes	PubChem , American Chemical Society
31	CH3(CH2)5OH	1-Hexanol (Hexan-1-ol)	C6H14O (or CH3(CH2)5OH)	Yes	PubChem , Wikipedia
32	(CH3)3SiOH	Trimethylsilanol	(CH3)3SiOH (or C3H10OSi)	Yes	Sigma-Aldrich , Wikipedia
33	((CH3)3Si)2O	Hexamethyldisiloxane	((CH3)3Si)2O (or C6H18OSi2)	Yes	Wikipedia , Thermo Fisher , Ereztech
34	CF3CONH2	Trifluoroacetamide	CF3CONH2 (or C2H2F3NO)	Yes	PubChem , Fisher Scientific

Note on $Fe(NH_4)_2SO_4$: The extracted formula $Fe(NH_4)_2SO_4$ does not directly correspond to the common compound Ferrous Ammonium Sulfate (Mohr's Salt), which is a double salt with the formula $(NH_4)_2Fe(SO_4)_2$ or its hexahydrate $(NH_4)_2Fe(SO_4)_2 \cdot 6H_2O$. The extracted formula suggests a 1:1 ratio of ammonium to sulfate, whereas Mohr's salt has a 2:2 (or 1:1 after simplifying the sulfate part if written as $FeSO_4 \cdot (NH_4)_2SO_4 \cdot 6H_2O$). The regenerated formula from the name "Ferrous ammonium sulfate" typically refers to the double salt. The PDF on page S-8 in the supporting information lists $Fe(NH_4)_2SO_4$ when discussing the quantitation of H_2O_2 . This might be a specific notation or a typo in the source document for what is commonly known as Mohr's Salt, or a different, less common species.

The analysis involved extracting formulas exactly as they appeared in the PDF, using web searches to generate their common chemical names, and then using those names to regenerate chemical formulas for comparison.