LLMs Got Rhyme? Hybrid Phonological Filtering for Greek Poetry Rhyme Detection and Generation

Stergios Chatzikyriakidis
Department of Philology
University of Crete
stergios.chatzikyriakidis@uoc.gr

Anastasia Natsina
Department of Philology
University of Crete
natsina@uoc.gr

Abstract

Large Language Models (LLMs), even though exhibiting multiple capabilities on many NLP tasks, struggle with phonologically-grounded phenomena like rhyme detection and generation. When one moves to lower-resource languages such as Modern Greek, this is even more evident. In this paper, we present a hybrid neural-symbolic system that combines LLMs with deterministic phonological algorithms to achieve accurate rhyme identification and generation. We implement a comprehensive taxonomy of Greek rhyme types and employ an agentic generation pipeline with phonological verification. We use multiple prompting strategies (zero-shot, few-shot, Chain-of-Thought, and RAG-augmented) across several LLMs including Claude 3.7 and 4.5, GPT-4o, Gemini 2.0 and open-weight models like Llama 3.1 8B and 70B and Mistral Large. Results reveal a significant reasoning gap: while native-like models (Claude 3.7) perform intuitively (40% accuracy in identification), reasoning-heavy models (Claude 4.5) achieve state-of-the-art performance (54%) only when prompted with Chain-of-Thought. Most critically, pure LLM generation fails significantly (under 4% valid poems), while our hybrid verification loop restores performance to 73.1%. Along with the system presented, we further release a corpus of 40,000+ rhymes, derived from the *Anemoskala* and *Interwar Poetry* corpora, to support future research.[1]

1 Introduction

Rhyme is a significant feature in many poetic traditions. In Modern Greek, rhyme (ομοιοκαταληξία) has been systematically employed from the Cretan Renaissance (14th–17th centuries) all the way to contemporary poetry and popular music, including hip-hop and rap (Topintzi et al., 2019).

LLMs, unsurprisingly given their prevalent text-based training, exhibit notable weaknesses in phonologically-aware reasoning. LLMs process text at the token level, which poorly aligns with phonological units like syllables, stress patterns, and rhyme domains.

We tackle the issue of rhyme identification and generation using a hybrid Neural-Symbolic architecture that combines the generative and reasoning capabilities of LLMs with deterministic phonological algorithms. Our contributions are:

1. A dataset of 40k rhymes aggregated from existing Modern Greek corpora, based on the phonological taxonomy found in Topintzi et al. (2019).

2. A hybrid proposer-verifier identification pipeline where LLM predictions are grounded by deterministic phonological verification.

3. An agentic generation pipeline with a Generate-Verify-Refine loop that raises the validity of generated poems from nearly 0% to 73.1%.

4. A multi-model evaluation across proprietary (Claude, GPT-4o) and open-weight models, with analysis of prompting strategies including RAG augmentation.

2 Background

2.1 Rhyme in Modern Greek Poetry

Rhyme in Modern Greek poetry dates to the medieval period, appearing in works like those of Stefanos Sachlikis (14th c.), and flourishing in Cretan Renaissance masterpieces such as Kornaros's *Erotokritos*. The phenomenon spans the Heptanesian Romanticism of Solomos, the Athenian school, Parnassian and Symbolist movements, and continues, albeit more selectively, in modern and contemporary poetry (Κοκόλης, 1993).

[1]The material needed to run the experiments and verify the results of this paper can be found here: https://github.com/StergiosCha/Greek_Rhyme

Proceedings of SIGHUM 2026, pages 87–101
March 28-29, 2026

Greek rhyme exhibits several distinctive characteristics:

Stress-based Classification Greek is a stress-accent language where rhyme domains are defined relative to the stressed syllable. More particularly, Greek is subject to the three-syllable rule according to which accent must fall in one of the last three syllables of the word (broadly defined, includes also cases of what we call phonological word, e.g. a word plus a weak object pronoun). We follow Topintzi et al. (2019) and use the following categories:

- **Masculine (M)**: Stress is on the final syllable (oxytone), e.g., καρδιά / φωτιά [/karˈðja / foˈtɕa/].
- **Feminine (F2)**: Stress is on the penultimate syllable (paroxytone), e.g., ελπίδα / πατρίδα [/elˈpiða / paˈtriða/].
- **Proparoxytone (F3)**: Stress is on the antepenultimate syllable, e.g., στόματα / σώματα [/ ˈstomata / ˈsomata/].

Greek poetry further employs several other rhyme types that are independent of stress position:

Rich Rhyme (RICH): In this type, the onset consonant(s) of the stressed syllable must match. This is further distinguished into Total Rich (TR) rhyme where complete onset matching is at play, or Partial Rich (PR) with partial matching:

- TR-S (singleton): καλά / μαλά [/kaˈla / maˈla/]
- PR-C1 (first consonant): στόματα / σώματα [/ ˈstomata / ˈsomata/]

Identical Pre-rhyme Vowel (IDV): The vowel that precedes the stressed syllable must match:

- ξανθή / γραφή [/ksanˈθi / ɣraˈfi/] (the pre-stress vowel /a/ matches).

Mosaic (MOS): The rhyme domain spans across more than one word:

- όνομά της / ο μπάτης [/ ˈonoma tis / o ˈbatis/]

Imperfect (IMP): Partial phonetic matching with systematic variation:

- IMP-V: Vowel differs (χάνετε / γίνετε [/ ˈxanete / ˈjinete/])
- IMP-C: Consonant differs (ξαφνίζει / τεχνίτη [/ ksafˈnizi / teˈxniti/])
- IMP-0F: Final consonant-zero alternation (πιστοί / χαρείς [/ piˈsti / xaˈris/])

2.2 Computational Approaches to Rhyme

Early approaches on rhyme identification used classic unsupervised machine learning (Reddy and Knight, 2011) or probabilistic models using phoneme frequencies for rhyme detection in rap music (Hirjee and Brown, 2010). More recent supervised methods based on neural networks achieve higher accuracies. For example, Haider and Kuhn (2018) achieve 97% accuracy via a single Siamese Recurrent Network model trained in German, English, and French, using no explicit phonetic features.

In poetry generation, we find early statistical machine translation for Classical Chinese quatrain generation by He et al. (2012), where they in effect treat each line as a kind of translation of the previous line. The Hafez system (Ghazvininejad et al., 2016) is a hybrid system that puts together finite-state acceptors that encode metrical and rhyme constraints with RNNs for English sonnet generation. Lau et al. (2018) develop Deep-speare, which is a joint neural model for Shakespearean sonnets. The outputs of this system have been shown to be to a large extent indistinguishable from human verse in crowd evaluations. More recent work has moved towards modeling rhyme as continuous similarity rather than categorical classes (e.g., Nagy, 2022, for Latin), and unsupervised detection through tools like RhymeTagger (Plecháč, 2018). Moving on to the LLM era, we find byte-level transformers such as ByGPT5 (Belouadi and Eger, 2023), as well as synthetic-data approaches like GPoeT (Popescu-Belis et al., 2023).

For Greek specifically, the foundational computational work is that of Topintzi et al. (2019), who established the categorical taxonomy for Greek rhyme and developed an initial *Greek Rhyme* (GrRh) database. While their work provided the theoretical framework and the original corpus curation, our study introduces several critical advancements: (i) the first full algorithmic implementation of the Greek-specific G2P and rhyming rules in a deterministic Python environment; (ii) a large-scale reproduction and cleaning of the corpus, using our verifier to identify and correct systematic extraction errors in the original GrRh, thereby recover-

Poet	Rhyme Pairs
Palamas	20,620
Tellos Agras	6,119
Valaoritis	4,202
Solomos	2,518
Karyotakis	1,692
Cavafy	1,585
Fotos Giofyllis	1,565
Kostas Ouranis	792
Napoleon Lapathiotis	495
Romos Filiras	484
Mitsos Papanikolaou	404
Kalvos	100
Total	**40,576**

Table 1: Corpus composition by poet.

ing over 40,000 high-quality rhyme pairs; and (iii) the integration of these rules into an agentic LLM framework for generation and "proposer-verifier" identification.

3 Dataset

We construct our dataset by using two primary resources for Modern Greek poetry. The first source is the Anemoskala archive from the Centre for the Greek Language (KEG - Κέντρο Ελληνικής Γλώσσας), which provides extensive digitized collections of major poets. The second source is the Interwar Poetry Dataset, an open-access dataset created by Dr Natsina and Professor Chatzikyriakidis with the help of undergraduate students at the Philology Department, University of Crete[2] This corpus comprises over 600 poems by interwar Greek poets. We apply phonological filtering to extract rhyme pairs from both corpora and then merge them. The corpus statistics are shown in Table 1.

4 System Architecture

We implement a hybrid, neural-symbolic architecture that combines LLMs with symbolic phonological rules (Figure 1). The aim is to tackle the inherent "blindness" of sub-token processing by delegating the phonological validation to a symbolic engine.

4.1 Phonological Engine

The basis of the phonological symbolic processor/verifier is based on (Topintzi et al., 2019). Its purpose is exactly to handle Greek-specific rhyme analysis:

- **Syllabification**. A number of Greek syllabification rules are implemented. The algorithm is designed to identify syllable boundaries based on Greek-specific phonotactic constraints.
- **Stress Detection**. We use the accent marks found in Greek orthography (ά, έ, ή, ί, ό, ύ, ώ) in order to identify stress. In case of words with clitics (notably weak object pronouns, e.g., χάλεσέ με), we have a mechanism to handle stress domain extension.
- **Rhyme Domain Extraction**. The domain of the rhyme extends from the stressed vowel until the end of the phonological phrase. In the case of Mosaic rhymes, it can span multiple orthographic words.
- **Phonetic Transcription**. We convert Greek orthography to a phonetic representation.

4.2 Rhyme Classification Module

We develop a classification module based on the phonological module. It compares rhyme domain pairs and assigns labels. More specifically it checks:

1. **Position Classification**: Here it determines where the stress falls and classifies as M/F2/F3.
2. **Perfect Match Check**: In this part, the post-stress material is checked to see if it is an exact match.
3. **Feature Detection**: This checks for RICH rhyme (onset matching), IDV rhyme (pre-stress vowel) and/or Mosaic (MOS) rhyme (word boundary crossing).
4. **Imperfection Analysis**: In case the rhyme is not perfect, classify accordingly (IMP-V, IMP-C, IMP-0F, IMP-0M).

The output result is a compound label such as `F2-TR-S-IDV` (Feminine-2, Total Rich Singleton, Identical pre-rhyme Vowel).

4.3 LLM Integration and Prompting Strategies

We evaluate the performance of several state-of-the-art LLMs (Claude 3.7/4.5, GPT-4o, Gemini 2.0) using a progressive series of prompting strategies: Zero-Shot Structured, Chain-of-Thought (CoT), and RAG-Augmented.

[2]Dataset available at: https://github.com/StergiosChatzikyriakidis/Modern_Greek_Literature/tree/v1.

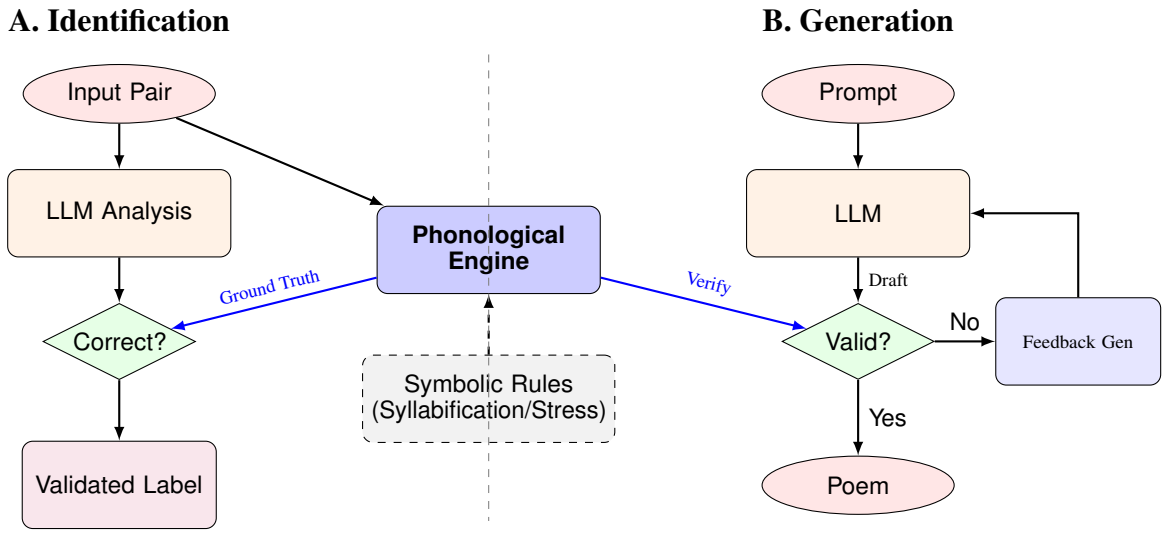

Figure 1: Hybrid system architecture. **Left**: Identification combines LLM predictions with Engine-generated ground truth for validation. **Right**: Generation uses the Engine to verify and refine LLM outputs.

Hybrid Combination Mechanism For the identification task, our system utilizes the LLM as a *proposer* and the symbolic engine as a *verifier*. In the "Verified Detection" mode, the system first obtains an initial classification from the LLM. We note here that the phonological ground truth for the validation is provided by the symbolic engine. The LLM receives this as prompt and is asked to *self-reflect* and compare its initial hypothesis with the deterministic rules and correct its final output, if needed. This ensures that the linguistic reasoning of the transformer is grounded in the deterministic accuracy of the phonological engine.

RAG Implementation Details Our RAG component (*Rhyme-RAG*) retrieves relevant examples from our cleaned 40,000-rhyme corpus. The retrieval we use is feature-based in the following sense: the system extracts phonological features (e.g., "F3", "RICH", "MOSAIC") from the user query via keyword markers and retrieves the top-15 most relevant rhyme pairs that exhibit these exact patterns. This provides the model with high-precision examples of the specific, often obscure, Greek rhyme types it is tasked to generate or identify.

4.4 Agentic Generation Pipeline

For rhyme generation, we use a Generate-Verify-Refine loop. The agent first generates a draft stanza based on a theme, requested rhyme pattern and number of lines. The symbolic part of the system is then tasked with identifying final words for every line and verifying them using symbolic rules. If a rhyme fails, the verifier generates a precise error message, which is sent back to the LLM for correction.

Algorithm 1 Generate-Verify-Refine Loop

Input: Theme, rhyme_type, features, num_lines
Output: Phonologically valid poem
$attempts \leftarrow 0$
while $attempts < 15$ **do**
  $poem \leftarrow LLM.generate(prompt)$
  $errors \leftarrow verify\_rhymes(poem)$
  if $errors = \emptyset$ **then**
    return $poem$
  else
    $feedback \leftarrow format\_errors(errors)$
    $prompt \leftarrow update\_prompt(feedback)$
    $attempts \leftarrow attempts + 1$
  end if
end while
return $poem$ with warning

Verifier Feedback Strings Concrete examples of these feedback strings include:

- `Stress mismatch: Expected F3, found F2 for 'αναβιώνει'/'απλώνει'.`
- `Identical words 'αγαπώ' / 'αγαπώ' (COPY) are not allowed unless requested.`
- `RICH requested but 'σκληρό' / 'θησαυρό' is PURE.`

This iterative feedback continues for up to 15 attempts.

The verifier's task is to check each rhyme pair using the phonological rules and provide feedback in case the rhyme does not comply with the rules.

5 Experimental Setup

We use a test set of 40 poems for rhyme identification. This produces a total of 160 test cases in total (2 strategies × 2 RAG configs × 40 poems). The set has a balanced distribution across rhyme types (13 Masculine, 16 Feminine (F2), 11 Proparoxytone (F3)), as well as comprehensive coverage of rhyme features including more rare types of rhyme (21 PURE, 10 RICH, 10 IDV, 6 IMPERFECT, 5 MOSAIC). We went for a balance that was big enough to make meaningful claims, while, at the same time, maintaining evaluation feasibility across 8 models and 4 configurations (1,280 total API calls).

For generation, we evaluate the system on 26 test cases with specified rhyme constraints (e.g., "Write a 4-line poem with F3-RICH rhyme on theme: love"). To calculate the "Generation Validity (%)" metric, we evaluate the LLM output against the requested rhyme scheme. A poem is considered valid only if **all** pairs in the requested scheme (e.g., A-A and B-B in a quatrain) are verified as correct matches by the symbolic engine. If a model generates three lines of monorhyme (AAA) when AABB was requested, it is penalized for the structural mismatch. Each test runs twice: once with our verification loop (Generate-Verify-Refine) and once without (pure LLM generation).

6 Results

6.1 Rhyme Identification Results

In Table 2, we see the results across all prompting configurations.

Table 3 breaks down performance by rhyme type, revealing systematic biases.

The results across all configurations (Table 2) show that Claude 4.5 reaches the highest accuracy with 53.8%, but this number comes with a caveat: it requires both CoT and RAG to get there. GPT-4o goes from 7.7% to 50.0% once CoT is switched on, the model knows more than it lets on, but needs to be walked through the reasoning step by step. Claude 3.7 sits around 38-42% regardless of what you throw at it.

The picture we get from open models is rather poor. Llama 3.1 70B tops out at 23.1% and CoT makes things worse for Llama 3.3 (down to 7.7%). This is not just about size. If it were, CoT should help the 70B models the way it helps GPT-4o. It does not, which means the Greek phonetic representations in these models are simply not there to reason over.

Model size helps, the 70B Llama shows better performance than the 8B easily, but it does not close the gap with the proprietary systems.

Analyzing performance by rhyme type (Table 3) exposes systematic biases prevalent across the LLM landscape. Proparoxytone (F3) rhymes represent a clear ceiling for all models, with accuracy dropping significantly compared to Masculine (M) rhymes. This performance decay is likely due to the increased distance between the stress and the word boundary, which multiplies the potential for syllabification and stress-mapping errors during sub-token processing.

Most critically, complex features like RICH or MOSAIC are almost entirely invisible to current transformers (see Appendix for full statistics); MOSAIC detection achieved 0% accuracy across almost all configurations. This reinforces our central argument: LLMs, while creatively powerful, lack the precision required for dealing with rhyme identification and generation.

While our proposed system architecture (Section 4) includes a verified detection mode where the LLM can reflect on symbolic ground truth, the benchmarks reported here measure the models' performance in unaided, CoT, and RAG configurations. This experimental design allows us to isolate the phonological reasoning gap and measure the inherent difficulty of the Greek rhyming taxonomy for neural architectures.

To understand the models' capabilities and failure modes, we categorized the 9 test cases into thematic observations (detailed outputs in Appendix A):

1. **Phonological vs. Lexical Conflation**: Many models struggled to distinguish phonemically identical but orthographically distinct pairs (homonyms) from simple repetition (COPY rhyme). For example, in the pair κρίνοι/κρίνει (lilies / it judges), which are perfect homophones, Claude 4.5 falsely flagged a 'COPY' rhyme, failing to recognize the distinct lemmata despite the sound identity.

2. **The Proparoxytone Distance Problem**: As seen in the quantitative results, proparoxytone rhymes like παράπονο/άπονο caused significant structural failures. Models often failed to map the three-syllable distance between the stress and the word boundary, frequently misidentifying these as paroxytone (F2) or simple masculine rhymes.

Model	Structured	Structured+RAG	CoT	CoT+RAG
Proprietary Models				
Claude 4.5	53.8%	26.9%	46.2%	**53.8%**
Claude 3.7	38.5%	**42.3%**	42.3%	30.8%
GPT-4o	7.7%	7.7%	**50.0%**	26.9%
Gemini 2.0	23.1%	**42.3%**	19.2%	11.5%
Open-Weight Models				
Mistral Large	11.5%	15.4%	**26.9%**	11.5%
Llama 3.1 70B	**23.1%**	19.2%	11.5%	15.4%
Llama 3.3 70B	**23.1%**	23.1%	7.7%	7.7%
Llama 3.1 8B	7.7%	15.4%	7.7%	3.8%

Table 2: Rhyme identification accuracy (%) across configurations. Bold means best configuration per model.

Model	M (n=32)	F2 (n=40)	F3 (n=32)
Claude 4.5	**65.6%**	47.5%	21.9%
Claude 3.7	37.5%	**52.5%**	21.9%
GPT-4o	31.2%	25.0%	12.5%
Gemini 2.0	43.8%	20.0%	9.4%
Mistral Large	25.0%	17.5%	6.2%
Llama 3.1 70B	12.5%	32.5%	3.1%
Llama 3.3 70B	25.0%	12.5%	9.4%
Llama 3.1 8B	6.2%	12.5%	6.2%

Table 3: Identification accuracy by rhyme type. All models struggle most with F3 (proparoxytone) rhymes.

Model	No Verify	With Verify	Improvement
Claude 4.5	0.0%	34.6%	+34.6%
Claude 3.7	3.8%	**73.1%**	+69.3%
GPT-4o	0.0%	42.3%	+42.3%

Table 4: Generation validity (% of perfectly valid poems) with and without phonological verification loop.

3. **Visual vs. Phonetic Identity**: A core challenge for sub-token transformers is getting tricked by orthography. Most models failed on the pair *ορθός/φως* due to the visual mismatch, yet reasoning-heavy configurations (Claude 4.5 CoT+RAG) were able to parse the underlying /os/ identity, suggesting that step-by-step reasoning can partially overcome the visual bias of the tokenizer.

4. **Morphological Variation**: Test cases involving archaic forms (e.g., the dative plural *ουρανοίς*) proved particularly challenging.

6.2 Rhyme Generation Results

Table 5 shows generation validity broken down by feature complexity.

The verification loop enhances rhyme validity across all models. Claude 3.7 achieves the highest verified generation rate (73.1% valid poems),

Feature Type	n	Claude 3.7	Claude 4.5	GPT-4o
BASIC (no features)	6	**100.0%**	83.3%	90.9%
IMPERFECT	2	50.0%	50.0%	33.3%
IDV+PURE	2	50.0%	0.0%	66.7%
IDV+RICH	2	50.0%	0.0%	0.0%
IDV+IMPERFECT	2	**100.0%**	0.0%	**100.0%**
IDV+MOSAIC+PURE	2	0.0%	0.0%	0.0%

Table 5: Generation validity (%) with verification, by feature complexity. BASIC rhymes (no features) achieve highest success. Complex multi-feature combinations (e.g., IDV+MOSAIC+PURE) fail even with verification. We note that the small sample sizes per feature category (n=2 for most non-BASIC types) limit the reliability of these breakdowns. As such, they are presented to illustrate qualitative trends rather than robust estimates.

demonstrating that the hybrid approach can be used in order to compensate for LLM phonological weaknesses. Pure LLM generation fails catastrophically (0-4% validity), confirming that deterministic verification is essential for constrained poetry generation, at least for the models used.

6.3 Qualitative Analysis of Generation

Three main types of generation errors (full output traces are provided in Appendix B) are generally corrected by the loop:

1. **Correcting the stress pattern (F3 vs F2)**: When asked for F3 (proparoxytone) rhymes, models are often incorrectly defaulting to F2 (paroxytone). *Example*: GPT-4o initially generated *αναβιώνει* / *απλώνει* (F2). The verifier returned `"Stress mismatch: Expected F3, found F2"`. The model corrected this to a valid F3 rhyme in the subsequent iteration.

2. **Feature Precision (Rich vs Pure)**: LLMs often treat rhyme types loosely. When M-PURE was requested, Claude 3.7 generated *σκληρό* / *θησαυρό* (which is RICH, sharing

the /r/ onset). After the intervention of the loop, the model successfully refined the output to a strict PURE rhyme, demonstrating the system's ability to enforce precise stylistic constraints.

3. **Non-rhymes**: In some cases, Claude 4.5 proposed non-rhyming outputs, e.g. pairs like αγαπώ / ξέρω. The verification loop noted these invalid pairs (`"No rhyme"`), and forced the model to regenerate valid phonological matches.

7 Discussion

7.1 Why LLMs Struggle with Rhyme

One issue is potentially tied to the nature of LLM tokenizers (e.g., BPE, WordPiece), i.e. the fact that they segment text based on statistical co-occurrence and not linguistically motivated units. For example, a word like παράπονο (/paˈrapono/) might be tokenized as [πα, ρά, πο, νο] in an ideal case, but often appears as [παρ, άπ, ονο] depending on the vocabulary, stripping the model of the ability to map the stress position (F3) relative to the final syllable.

Unlike orthographical systems where text maps close to 1:1 to sound, Greek's retention of historical orthography, i.e. a system of orthography that has been kept essentially the same for millennia and, thus, has not followed the changes in the language, features many-to-one mappings (e.g., the sound /i/ can be spelled as $\iota, \eta, \upsilon, \epsilon\iota, o\iota, \upsilon\iota$). LLMs trained primarily on text often rely on visual similarity rather than phonetic identity. This explains why models fail on *χρίνοι/χρίνει* (visual mismatch but phonetic match) while hallucinating rhymes for *ορθός/φως* (visual mismatch and phonetic mismatch).

Finally, note that Greek rhyme is strictly defined by stress position (M, F2, F3). Since LLMs lack an explicit prosodic module, they are prone to fail when attempting to distinguish minimal pairs that differ only in stress (e.g., νόμος vs νομός).

7.2 Creativity vs. Hallucination

A number of words that the LLM proposes are hallucinations, in the sense that these words are not existing words in Modern Greek, However, the analysis of these words shows a very nuanced trade-off between semantic grounding and poetic creativity. Two categories of invented vocabulary are observed.

Some of the hallucinations can be very well-taken as poetic neologism. For example, Claude 4.5 gives us αιωνημένα (eternal-ized) and θάρραμα (courage-thing), which are not only valid phonotactically, but also are meaningful enough neologisms (for example αιωνημένα can be an adjective that means something related to eternity, while θάρραμα can be seen as a collage of θάρρος (courage) and χάραμα (dawn). A striking example is *πυραφί* (fire-colored), which appears to be constructed by analogy to *χρυσαφί* (gold-colored), demonstrating a deep (if unauthorized) grasp of Greek morphology. Similarly, αγέρη appears as a valid homophone of αγέρι (breeze), suggesting a spelling variation rather than a failure. Phonetic failures, on the other hand, produce words that are phonotactically correct but lack semantic transparency. Examples include τσίγκλο and σκάμπουνε. It is significant that no phonotactic violations occur in these cases, pointing to the models' robust internal representation of Greek syllable structure. We find a distinct behavioral difference between models: GPT-4o is the safest model, producing almost zero nonsense but also fewer neologisms. In contrast, Claude 4.5 is the boldest, with a high rate of invention, the majority of which are plausible neologisms rather than nonsense. This suggests that what is often penalized as hallucination in factual tasks can serve as a proxy for creativity in poetic tasks.

Crucially, this higher level of invention coincides with stronger performance in the verification loop. It appears that the strict constraints imposed by the verifier push capable models to neologize in order to conform to the poetic form, inventing new words when standard vocabulary fails to meet the phonological requirements.

8 Future Work

The natural next step for such a system is to go beyond rhyme and implement metrical verification. Greek poetry relies heavily on stress-timed constituent meters (e.g., iambic 15-syllable verse) and systems that could handle both rhyme and metrical structure would be a very interesting research direction to take.

Furthermore, we plan to explore Reinforcement Learning from Phonological Feedback (RLPF). The idea is that instead of a simple rejection sampling loop at inference time, the signals from our deterministic verifier could be used as a reward function to fine-tune a smaller model (e.g., Llama

8B), potentially internalizing phonological constraints directly into the model's weights.

9 Conclusion

In this work, we presented a hybrid neuro-symbolic system for Modern Greek poetry identification and generation. We showed that while LLMs seem to possess latent creative capabilities, they struggle with the precision needed in order to deal with rhyme detection and generation. We combined a deterministic phonological engine with an agentic generation loop, and achieved a significant improvement in generation validity, raising success rates from a baseline of under 4% to 73.1%. Furthermore, our identification benchmarks revealed a reasoning gap, in which only the most advanced reasoning models (using Chain-of-Thought) could compete with symbolic verification. We hope that our released codebase and the curated corpus of 40,000+ Modern Greek rhymes will serve as foundational resources for future research into phonologically-aware NLP.

Limitations

There are a number of limitations to this work. The rhyme taxonomy we use is categorical, inherited from Topintzi et al. (2019). There is recent work arguing that rhyme is better treated as a gradient phenomenon (Nagy, 2022), and our deterministic engine has nothing to say about degrees of rhyme, i.e. a pair either matches or it does not. The corpus, while large by Greek standards, remains small next to what is available for English or German. On the generation side, passing the verifier means the rhymes are structurally correct, but says nothing about whether the poem is any good; phonological validity and poetic quality are orthogonal. Finally, every failed verification triggers a new LLM call, which means the loop can be substantially slower than ordinary single-pass generation.

Acknowledgments

We gratefully acknowledge the Centre for the Greek Language (Κέντρο Ελληνικής Γλώσσας) for providing access to the Anemoskala corpus and granting permission to derive our rhyming dataset from their digital resources. The original corpus is available through the Portal for the Greek Language (www.greek-language.gr).

References

Jonas Belouadi and Steffen Eger. 2023. ByGPT5: End-to-end style-conditioned poetry generation with token-free language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7364–7381, Toronto, Canada. Association for Computational Linguistics.

Marjan Ghazvininejad, Yejin Choi, and Kevin Knight. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191.

Thomas Haider and Jonas Kuhn. 2018. Supervised rhyme detection with siamese recurrent networks. In *Proceedings of the Workshop on Stylistic Variation*, pages 81–86.

Jing He, Ming Zhou, and Long Jiang. 2012. Generating chinese classical poems with statistical machine translation models. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1650–1656.

Hussein Hirjee and Daniel Brown. 2010. Automatic detection of rhyme in rap music. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 395–400.

Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1948–1958, Melbourne, Australia. Association for Computational Linguistics.

Ben Nagy. 2022. Rhyme in classical latin poetry: stylistic or stochastic? *Digital Scholarship in the Humanities*, 37(4):1097–1118.

Petr Plecháč. 2018. A collocation-driven method of discovering rhymes (in czech, english, and french poetry). In *Taming the Corpus: From Inflection and Lexis to Interpretation*, pages 79–95. Springer.

Andrei Popescu-Belis, Àlex R. Atrio, Bastien Bernath, Etienne Boisson, Teo Ferrari, Xavier Theimer-Lienhard, and Giorgos Vernikos. 2023. GPoeT: a language model trained for rhyme generation on synthetic data. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 10–20, Dubrovnik, Croatia. Association for Computational Linguistics.

Sravana Reddy and Kevin Knight. 2011. Unsupervised discovery of rhyme schemes. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 77–82.

Nina Topintzi, Konstantinos Avdelidis, and Thomai Valkanou. 2019. Quantifying greek rhyme. In *Selected Papers from the 23rd International Symposium on Theoretical and Applied Linguistics*, pages 429–447. School of English, Aristotle University of Thessaloniki.

Ξενοφών Κοκόλης. 1993. Η ομοιοκαταληξία: Τύποι και λειτουργικές διαστάσεις. Στιγμή, Athens.

A Detailed Qualitative Results

This appendix presents the full raw outputs for the 9 representative test cases discussed in the Qualitative Analysis. For each case, we show the predicted Rhyme Type and Features for all 8 models across 4 configurations (Structured vs CoT, No RAG vs RAG).

Legend: ✓ = Correct Type & Features, ☞ = Correct Type but Wrong Features, ✗ = Incorrect Type.

A.1 1. Baseline Success (M) (Keyword: 'απαιτώ')

Poem: Έτσι από σένα περιμένω κι απαιτώ. / της Τραγωδίας τον Λόγο τον λαμπρό —
True Label: M ['PURE']

Model	Struct (No RAG)	Struct (RAG)	CoT (No RAG)	CoT (RAG)
Claude 4.5	☞ M	✓ M [PURE]	✗ S	☞ M
Claude 3.7	✗ MISS	✗ MISS	☞ M	✗ I
GPT-4o	✗ MISS	☞ M [IMPERFECT]	☞ M	✗ S
Gemini 2.0	☞ M [IMPERFECT]	☞ M [IMPERFECT]	✗ MISS	✗ MISS
Llama 70B	☞ M [IDV, RICH]	✗ F2 [PURE]	✗ F2	✗ +
Llama 3.3	✗ S	☞ M [IMPERFECT]	✗ MISS	✗ OF
Mistral	☞ M [IMPERFECT]	✗ STRESS	✗ S	✗ IN
Llama 8B	✗ S	✗ MISS	✗ WORKERJOE	✗ ANN

A.2 2. Baseline Success (F2) (Keyword: 'κρίνοι')

Poem: Από ρουμπίνια ρόδα, από μαργαριτάρια κρίνοι, / από αμεθύστους μενεξέδες. Ως αυτός τα κρίνει,
True Label: F2 ['RICH']

Model	Struct (No RAG)	Struct (RAG)	CoT (No RAG)	CoT (RAG)
Claude 4.5	☞ F2 [COPY, RICH]	✗ BETWEEN	✗ S	☞ F2
Claude 3.7	☞ F2 [COPY]	☞ F2 [COPY]	✗ M	✗ TR [CC]
GPT-4o	✗ S	✗ S	☞ F2	✗ MASCULINE
Gemini 2.0	✗ MISS	✗ MISS	✗ S	✗ MISS
Llama 70B	☞ F2 [2W, C2, IDV, RICH]	✓ F2 [RICH]	✗ S	✗ TR
Llama 3.3	✗ MISS	☞ F2 [PURE]	✗ M	✗ MISS
Mistral	✗ STRESS	☞ F2 [CC, RICH]	✗ S	✗ 2
Llama 8B	✗ S	✗ RECEIVE	✗ CLIENT	✗ M

A.3 3. Structural Failure (F3) (Keyword: 'παράπονο')

Poem: αρχίζει το μωρό ένα παράπονο, / που ήρθε σ' έναν κόσμο τόσον άπονο.
True Label: F3 ['PURE']

Model	Struct (No RAG)	Struct (RAG)	CoT (No RAG)	CoT (RAG)
Claude 4.5	✗ MISS	✗ IN	✗ S	✗ NOT
Claude 3.7	✗ MISS	☞ F3 [MOSAIC]	☞ F3	☞ F3 [IMPERFECT]
GPT-4o	✗ MISS	✗ MISS	✗ MATCH	☞ F3 [IMPERFECT, RICH]
Gemini 2.0	✗ MISS	☞ F3 [IDV, RICH]	✗ MISS	✗ F2
Llama 70B	✗ MISS	✗ F2 [PURE]	✗ F2	✗ M
Llama 3.3	✗ M [C1, RICH]	✗ MISS	✗ S	✗ MISS
Mistral	✗ STRESS	✗ F2 [IDV, PURE]	✗ S	✗ MISS
Llama 8B	✗ K	✗ F2 [IDV, RICH]	✗ S	✗ ZIGSPACE

A.4 4. Mosaic Failure (Keyword: 'λυγμέ')

Poem: του αθηναίικου εσύ χινόπωρου λυγμέ, / ψιχάλα κυνηγάρα, που έβρεχες για με;
True Label: M ['MOSAIC']

Model	Struct (No RAG)	Struct (RAG)	CoT (No RAG)	CoT (RAG)
Claude 4.5	☞ M [COPY]	☞ M [COPY]	☞ M [COPY]	☞ M [MOSAIC, PURE]
Claude 3.7	☞ M [IMPERFECT]	✓ M [MOSAIC]	✗ S	☞ M [PURE]
GPT-4o	✗ S	✗ MISS	☞ M [COPY]	☞ M
Gemini 2.0	✗ MISS	☞ M [IMPERFECT]	✗ MISS	☞ M
Llama 70B	✗ S	✗ MISS	✗ S	✗ NO
Llama 3.3	✗ GIVEN	☞ M [IMPERFECT]	✗ S	✗ MISS
Mistral	✗ STRESS	✗ STRESS	☞ M [IMPERFECT, RICH]	☞ M [IMPERFECT]
Llama 8B	☞ M	✗ MISS	✗ SATURDAY	✗ S

A.5 5. Feature Hallucination (Keyword: 'αφρός')

Poem: Γιασεμιά, και κοράκια. Και των άσπρων ο αφρός / και του μαύρου η φοβέρα πάντα εντός μου κι εμπρός.
True Label: M ['PURE']

Model	Struct (No RAG)	Struct (RAG)	CoT (No RAG)	CoT (RAG)
Claude 4.5	☞ M [IMPERFECT]	☞ M [C1, RICH]	☞ M	✓ M [PURE]
Claude 3.7	✗ S	☞ M [RICH]	☞ M	✗ IN
GPT-4o	✗ S	✗ IN	☞ M	✗ IN
Gemini 2.0	☞ M [C2, RICH]	✓ M [PURE]	☞ M	✗ S
Llama 70B	☞ M [IMPERFECT]	✗ MISS	✗ [IA]	✗ F2
Llama 3.3	✗ S	✗ OF	✗ IN	☞ M
Mistral	✗ STRESS	✗ MISS	☞ M [F2, IMPERFECT]	☞ M [IMPERFECT]
Llama 8B	✗ MISS	✗ F2 [MOSAIC]	✗ STRING	✗ COMPLETE

A.6 6. Archaic Failure (Keyword: 'Ελληνίς')

Poem: Την εγέννησεν είς δήμος, μία πόλις Ελληνίς, / αλλ' ευθύς εκείνη έπτη, κι έστησεν εν ουρανοίς
True Label: M ['RICH']

Model	Struct (No RAG)	Struct (RAG)	CoT (No RAG)	CoT (RAG)
Claude 4.5	✗ MISS	✗ S	☞ M	☞ M [PURE]
Claude 3.7	✗ F2 [RICH]	☞ M [PURE]	✗ S	☞ M
GPT-4o	✗ OF	✗ OF	☞ M [IDENTICAL]	✗ S
Gemini 2.0	☞ M [IMPERFECT]	✗ MISS	☞ M	✗ MISS
Llama 70B	✗ ING	✗ MISS	☞ M	✗ NO
Llama 3.3	✓ M [RICH]	✗ F2 [PURE]	✗ BOTH	✗ OF
Mistral	✗ MISS	✗ F2 [C1, IMPERFECT, RICH]	✗ S	✗ MISS
Llama 8B	✗ IRSECURITY	✗ DAV	✗ CHOOSING	✗ MISS

A.7 7. Imperfect Detection (Keyword: 'γρήγορο')

Poem: αόριστη, με διάβα γρήγορο, / Στου καφενείου την είσοδο
True Label: F3 ['IMP', 'C', 'IMPERFECT']

Model	Struct (No RAG)	Struct (RAG)	CoT (No RAG)	CoT (RAG)
Claude 4.5	✗ S	✗ S	☞ F3 [IMPERFECT]	✗ IMPERFECT
Claude 3.7	☞ F3 [C1, IMPERFECT, RICH]	✗ S	✗ F2	✗ I
GPT-4o	☞ F3 [IMPERFECT]	✗ F2 [IMPERFECT]	✗ F2 [F2]	✗ S
Gemini 2.0	✗ MISS	☞ F3 [IMPERFECT]	✗ MISS	✗ F2
Llama 70B	✗ MISS	✗ F2 [IMPERFECT]	✗ M [IMPERFECT]	✗ S
Llama 3.3	✗ MISS	✗ M [IMPERFECT]	✗ MISS	✗ MISS
Mistral	✗ STRESS	✗ STRESS	✗ F2 [IMPERFECT]	✗ STRESS
Llama 8B	✗ F2 [2W, IDV]	✗ MISS	✗ D	✗ S

A.8 8. Proper Noun Failure (Keyword: 'νιότα')

Poem: Πολεμιστή, τα γαληνά σου νιότα / Όπου έπεσες, κλωνάρια κι απ' του Ευρώτα
True Label: F2 ['PURE']

Model	Struct (No RAG)	Struct (RAG)	CoT (No RAG)	CoT (RAG)
Claude 4.5	☞ F2	✗ STEP	☞ F2	☞ F2
Claude 3.7	☞ F2 [IMPERFECT]	✗ S	✗ F3	✗ I
GPT-4o	✗ S	✗ IN	☞ F2	✗ M [IDV, PURE]
Gemini 2.0	✗ MISS	✗ MISS	☞ F2	☞ F2
Llama 70B	✗ MISS	✗ MISS	☞ F2	☞ F2
Llama 3.3	✗ MISS	✗ THE	✗ SINCE	✗ BUT
Mistral	✗ STRESS	✗ MISS	✗ S	✗ 3
Llama 8B	✗ S	✗ F3 [2W, CC, IDV, RICH]	✗ THE	✗ F3

A.9 9. Visual vs Phonetic (Keyword: 'ορθός')

Poem: άξαφνα το παράθυρο και στάθηκα ορθός, / τις μυρωδιές, τα χρώματα και το ιλαρό το φως.
True Label: M ['MOSAIC', 'IDV']

Model	Struct (No RAG)	Struct (RAG)	CoT (No RAG)	CoT (RAG)
Claude 4.5	☞ M [COPY]	☞ M [IMPERFECT]	☞ M [IDV, IMPERFECT]	☞ M [PURE]
Claude 3.7	✗ MISS	☞ M [IMPERFECT]	✗ I	✗ IN
GPT-4o	✗ S	✗ S	✗ MISS	☞ M [PURE]
Gemini 2.0	✗ MISS	☞ M [IMPERFECT]	✗ MISS	✗ MISS
Llama 70B	✗ S	✗ MISS	✗ S	☞ M [IMPERFECT]
Llama 3.3	☞ M [IMPERFECT]	☞ M [IMPERFECT]	✗ MISS	✗ IN
Mistral	✗ MISS	☞ M [PURE]	✗ S	✗ ONSET
Llama 8B	✗ MISS	✗ THE	✗ BETWEEN	✗ S

B Appendix: Detailed Generation Traces

We present full interaction traces for the three classes of generation errors discussed in the Qualitative Analysis. These examples demonstrate how the hybrid verification loop intercepts compliant but incorrect outputs (Stress, Feature Precision) and hallucinations, guiding the model to phonologically valid results.

B.1 1. Stress Pattern Correction (GPT-4o)

Test Case: Theme "Sky" (*ουρανός*), Constraint: **F3 (Proparoxytone) Rhyme**.
The model initially defaults to the more common F2 pattern.
Initial Draft (Pure LLM):

> Στο *χέρι σου* το *μυστικό* *αναβιώνει*,
> *και* μέσα στα *σκιερά* *πελάγη* *απλώνει*.

Verifier Feedback: `Stress mismatch: Expected F3, found F2 for 'αναβιώνει'/'απλώνει'.`
Refined Output (Hybrid):

> Ω *θεέ μου*, στον *ουρανό* να *ψιθυρίσουμε*,
> τα *όνειρά* μας να *πετάξουν* να *αγγίξουμε*.

Analysis: The corrected rhymes (*ψιθυρίσουμε* / *αγγίξουμε*) correctly carry stress on the antepenultimate syllable.

B.2 2. Feature Precision (Claude 3.7)

Test Case: Theme "Light" (*φως*), Constraint: **M-PURE Rhyme**.
The model generates a valid rhyme that violates the "Pure" constraint (by being "Rich").
Initial Draft (Pure LLM):

> Θα *λάμπει* *πάντα* *φωτεινό* στον *κόσμο* το *σκληρό*,
> *γλυκαίνει* *κάθε* *σκοτεινή* *γωνιά* σαν *θησαυρό*.

Verifier Feedback: `PURE requested but 'σκληρό'/'θησαυρό' is RICH (Onset: /r/).`
Refined Output (Hybrid):

> Το *φως* σαν *κύμα* της *χαράς* *ξυπνά*
> τα *σκοτεινά* *μου* *όνειρα* *φωτίζει* *καθαρά*

Analysis: The refined pair (*ξυπνά* / *καθαρά*) has distinct onsets (/p/ vs /th/), satisfying the PURE constraint.

B.3 3. Hallucination Correction (Claude 4.5)

Test Case: Theme "Love" (*αγάπη*), Constraint: **M Rhyme**.
The model hallucinates rhymes with no phonetic overlap.
Initial Draft (Pure LLM):

> Η *καρδιά* *μου* *χτυπά* *δυνατά* *για* *σένα* *που* *αγαπώ*,
> *κάθε* *στιγμή* *που* *περνά* *χωρίς* *εσένα* *είναι* *πικρή*, το *ξέρω*.

Verifier Feedback: `No rhyme: 'αγαπώ' / 'ξέρω'.`
Refined Output (Hybrid):

> Μέσα στην *καρδιά* *μου* *κρύβεται* *αγαπώ*,
> *ένα* *όνειρο* *γλυκό* *που* *με* *κρατώ*,

Analysis: The model successfully regenerates a valid rhyme pair (*αγαπώ* / *κρατώ*).

C Appendix: Granular Feature Detection Statistics

This section provides the detailed identification accuracy for specific rhyme features (RICH, MOSAIC, IDV, IMPERFECT). These features represent the most challenging phonological patterns for current LLMs.

Model	PURE (n=68)	RICH (n=24)	MOSAIC (n=4)	IDV (n=24)	IMP (n=8)
Mistral Large	4.4%	**20.8%**	0.0%	12.5%	12.5%
Llama 3.1 70B	5.9%	16.7%	0.0%	12.5%	**37.5%**
Claude 3.7	8.8%	12.5%	0.0%	0.0%	25.0%
Claude 4.5	5.9%	8.3%	0.0%	4.2%	25.0%
Llama 3.3 70B	2.9%	8.3%	0.0%	0.0%	**37.5%**
Gemini 2.0	5.9%	0.0%	0.0%	0.0%	**37.5%**
GPT-4o	5.9%	0.0%	0.0%	0.0%	25.0%

Table 6: Feature detection accuracy (%) by individual feature type across all models.

D Representative Prompt Template

We provide the system prompt used for the **Zero-Shot Structured** identification task. This template establishes the phonological taxonomy and the structured output format as it was provided to the LLMs.

```
You are a Greek poetry rhyme analyzer. Your task is to identify rhyme patterns in Modern
Greek poetry.
```

RHYME TAXONOMY:

1. **Position-based Classification:**
   - M (Masculine): Rhyme from final stressed vowel to line end
   - F2 (Feminine-2): Rhyme from penultimate stressed vowel to line end
   - F3 (Feminine-3)[3]: Rhyme from antepenultimate stressed vowel to line end
2. **Complexity Features:**
   - **RICH (onset consonants match)**:
     - TR-S: Total rich with singleton onset (καλά - ξαλά)
     - TR-CC: Total rich with complex onset (αϋγή - ναϋγή)
     - PR-C1: Partial rich, first consonant matches (στόματα - σώματα)
     - PR-C2: Partial rich, second consonant matches (φοβερίζουν - τρίζουν)
   - **IDV (Pre-rhyme vowel identity)**: Vowel before stressed syllable matches.
   - **MOS (Mosaic)**: Rhyme domain crosses word boundaries.
   - **IMP (Imperfect)**:
     - IMP-V: Stressed vowel differs (χάνετε - γίνετε)
     - IMP-C: Consonants differ (ξαφνίζει - τεχνίτη)
     - IMP-0F: Final consonant-zero alternation
     - IMP-0M: Medial consonant-zero alternation
   - **COPY**: Complete word/phrase repetition.

ANALYSIS PROCEDURE: 1. Identify line-final stress position. 2. Extract rhyme domain (from stressed vowel rightward). 3. Compare with other lines (scan 4-line window by default). 4. Classify position (M/F2/F3) first. 5. Then identify features (RICH, IDV, MOS, IMP, COPY). 6. Use phonetic similarity, NOT just orthography.

POEM TO ANALYZE: *[Input Text]*

[3]The prompt retains the original experimental term. In the paper body, we use the label "F3" (proparoxytone rhyme) following Topintzi et al. (2019).