

TrochAIC: Metrical Tools for AI Interpretability

Ben Glaser¹ 

¹ Sterling Library and Poorvu Center for Teaching and Learning, Yale University, New Haven, CT, USA

Abstract

Versification has not been central to studies of AI poetry and poetic style. If we build tools to study the prosodic and phonological capacities of AI, we learn not just about AI verse but about hidden histories of reception and genre. The computational humanities can unpack these histories and explain how LLMs reify a limited kind of poetry and language. We can also compile data to train models that behave differently. I share a >100K line data set of scanned 18th century iambic pentameter, compare its versification with generative AI verse, and train BERT and T5 models with this data to scan meter and classify complexity. This tool lays the groundwork for exposing text-based LLMs to phonology and poetic prosody.

Keywords: LLMs, phonology, prosody, poetics, transformers

1 Introduction

Literary critics, computer scientists, and tech marketers alike have raised the specter of LLM poetry as a measure of linguistic robustness, intelligence, and even humanity. Recently, Porter and Machery [11] published a highly visible and provocatively-titled essay “AI Generated Poetry is Indistinguishable from Human-Written Poetry and is Rated more Favorably,” with a computer scientists responding swiftly to observe their “staggering banality” [4].

Yet whether and how we find value in any poem is first a matter of literary theory and judgment. As Ryan Heuser succinctly puts it, “generative texts lack meaning not because they lack an author, but because they lack a context” [8]. Porter and Machery collapse poetry into whatever can be produced and count as “poetry” in a context of their choosing. This repeats that lack of context, or what Michele Elam calls LLM’s “algorithmic ahistoricity” [5].

I argue that when we build tools to study the prosodic and phonological capacities of AI, we learn that AI poetry is not quite ahistorical. Rather, the sound (or silence) within their model of language reveals hidden histories of reception and genre. Elam further conceptualizes poetry as anathema to machine learning because it “will not optimize.” In what follows I ask how we might optimize LLMs for the uncharacteristic task of metrical scansion. I then show how computational humanities lets us refine data and train models that behave differently—though I make no claim as to the value or desirability of AI-generated verse itself. The real value, I argue, is in adding new tools to historical and formalist methods so we can manifest how mainstream LLMs reify a limited kind of poetry and language and, more importantly, improve our sense of what human poetry was and is.

Ben Glaser. “TrochAIC: Metrical Tools for AI Interpretability.” In: *Computational Humanities Research 2025*, ed. by Taylor Arnold, Margherita Fantoli, and Ruben Ros. Vol. 3. Anthology of Computers and the Humanities. 2025, 1422–1438. <https://doi.org/10.63744/K9Mwiiszu7QL>.

© 2025 by the authors. Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0).

2 The Re-History of LLM Versification

LLM poetry incessantly rhyme even when asked not to, constructing symmetrical iambic tetrameter about limited themes. A perfect storm of digitization, copyright, literary history, and choice of training tasks makes even advanced models sound like the uninventive 1890's poets excoriated by modernists for heavily iambic rhythm. When asked to assess the value or function of meter, AI chatbots often use the adjective "sing-song," as do many undergraduates before they learn formal analysis. GPTs retrace and cement one limited trajectory of literary meter.

We might think of the history of poetic meter not simply as a game of freedom and constraint, but as an implicit knowledge of English phonology made explicit to varying degrees as an abstraction into rules or constraints. The performance of LLMs does not reflect this history or knowledge. In the early twentieth century, with the rise of free verse and modernist ideologies of poetry, the diverse metrical techniques and theories of the long-nineteenth century became regimented and then rejected.¹ This winnowing of prosodic genres was a critical inflection point not just for poetry, but for literary criticism. LLM poetry generation and analysis bears the imprint of this transformation of historical prosody and its place in literary analysis. The rise of large scale natural language processing over the past four years thus extends the twentieth century's reification of prosodic form. Prosody is viewed as and then becomes a mechanical task, and only a few such tasks turn out to be reproducible by AI.

3 LLM Poetry's Task Structure

McCoy et al. observe that Generative AI performs well on tasks they have either been trained to perform, or seen performed in data, but struggle to generalize from these [10]. McCoy gives the useful example of the internet-common letter cypher "Rot-13" (a → n, b → o, etc). LLMs can't generalize "Rot-14". A poetic analogue would be Haiku: LLMs can now be consistently prompted to produce a 5-7-5 syllabic pattern in English. Because counting syllables is an unusual and often difficult phonological task in English, and because LLMs do not "know" what a syllable is (without a separate program for counting), they cannot reliably create a 6-8-6 "Haiku+1," or any of the myriad esoteric syllabic stanzas of the past 100 years (2-4-6-8-2 cinquains being quite popular between 1910-1930, but not at the digitized volume of reddit's r/haiku).

It is unclear how LLMs can be trained to improve their syllabic intelligence. Daniel Von Strien has created a synthetic haiku dataset to perform DPO (Direct Preference Optimization) training of OpenHermes-2.5-Mistral-7b, creating "HaikuHermes."² Having been rewarded for producing 5 and 7 syllable chunks of text, this model is less likely to miscount syllables. Yet this is not a reproducible effort for prosodic form in general, as it relies heavily on easily scraped or generated synthetic data. More common is some form of instruction tuning. Porter and Machery's prompt for GPT 3.5 to "Write a short poem in the style of []" produces metrical quatrains with some evocative word choices. They give no discussion of alternate prompting strategies such as one or few-shot prompting or prompts that specify features of the poetry. Their singular prompt reflects a view of poetry as a genre of the proper name. Their study suggests that the public is not familiar with those genres (Dorothy Lasky is wonderful, but hardly a known style); they do however recognize poetry as a *prosodic* genre of rhyming quatrains as well as a lexicon. The study therefore shows that "human-out-of-the-loop" prompting for a "short poem" maps one prosodic genre, the quatrain, with the proper name providing themes. The result is less stylometrics than sociology. It suggests more about a popular poetry literacy learned by LLMs.

¹ [6]

² The training set and model can be viewed and deployed from <https://huggingface.co/davanstrien/HaikuHermes-0.1-7B>

Davis rejects the study on other grounds: AI’s “extremely limited technical toolbox.” Only AI-Shakespeare generates sonnet form, and AI-Dickinson alternates tetrameter and trimeter. Given that in neither case do the AI-poets produce anything like the target poet’s verse technique, but rather the “Shakespearian sonnet” and Dickinson’s most frequent hymnal stanza, this again confirms how LLMs succeed at tasks only when they align with easily abstracted prosodic genre. The closed loop of LLM training, prompting, and our assessment narrowly calibrates “poetry” into the prosodic genre of a vaguely English-American ballad stanza. If readers fail to see this narrowness, it is because they share a repertoire of prosodic genres restricted to gestures like the volta of a sonnet or the epigrammatic and Orientalized concision of the haiku.

One indicator of the kind of “instruction tuning” underpinning modern chatbot poetry is the dataset, method, and end-product “co-poet” designed by Chakrabarty et al. [2]. Setting aside whether this collaborative tool has value for human poetry generation, or what such a tool *should* look like, the project matters here because of the relationship between its *synthetic* dataset and *synthesized* instructions. The training data is open source; here are a few characteristic examples from the 800k+ line training set 1:

Task	Prompt	Response
“Start / End”	Generate a poetic sentence about ‘parodies’ and ending in ‘order’	Even in parodies ,Ä there was order
“Haiku”	Write a haiku that includes the word ‘Military volunteer’	Donkey engineer / Military volunteer / celebrates new year
“Rhyme”	Write a poetic sentence that ends in a word which rhymes with ‘raise’	The eyes that fix you in a formulated phrase
Simile / Metaphor	Generate a metaphor about ‘canvas’	The canvas is a patchwork

Table 1: Training data for “Co-Poet”

This training data format aligns the model with a composite approximation of some genre, superficially related to poetic genres. Aside from numerous local flaws in identifying poetic devices, what makes “co-poet” superficial at the macro level is the determination of genre by task, which attempts to remedy the model’s fundamental abstraction from genre to word embeddings. Just as important as this macro level are the tasks that it leaves out: training in meter, stanzas beyond couplets, syntax, slant rhyme, lineation, allusion and other forms of figurative language.

4 Formal Analysis of LLM Verse

The narrowness of instruction-tuned “poetry” crystallizes in two new interdisciplinary analyses presented at NLP and Computational Humanities conferences in late 2024. In “Sonnet or Not, Bot? Poetry Evaluation for Large Models and Datasets,” Walsh et al., develop a test for “how well LLMs can identify *poetic form* for more than 20 poetic forms and formal elements” [13].³ The study’s main conclusion is that a range of flagship LLMs classify “fixed poetic forms” better than “unfixed forms or formal elements.” The models struggle with longer “fixed forms” (villanelle, sestina, pantoums) compared to shorter (sonnet, ballad, haiku). The authors carefully note, however, that

³ They release a valuable database of annotated public domain poems with crucial metadata about where those poems appear in other training sets; this kind of work undoes the calculated opacity of much LLM training.

scholars in the digital humanities have long understood literary classification tasks as proof of the “fuzziness” or ambiguity of genre and formal concepts. Notably, the n=15 study of human classification attempts showed very poor performance, especially on a 13-line “haiku” that appears chosen despite lacking fixed syllable or line counts and not being identified by its author as a haiku.⁴

If both poetic genre and form are constitutively “fuzzy,”⁵ and if, as Walsh et al. acknowledge, we struggle to know “whose versions of form will be reproduced” by LLMs, it is crucial to reflect on our “domain expertise” before further training and evaluating this new linguistic domain. The bigger problem, familiar to scholars of historical prosody, is that “form” as such does not delimit a coherent domain. The concept of poetic form both precedes and induces LLM’s “algorithmic ahistoricity.”⁶

Walsh et al.’s second study, “Does ChatGPT have a Poetic Style,” seeks a more fine-grained analysis of form. Like the preceding study, they calibrate this analysis around the distinction between “fixed forms,” “formal elements,” and “unfixed forms” [14]. The latter are not forms, however, but various subgenres of lyric with varying prominence (there are 254 elegies but only 16 aubades in the dataset). Formal “elements” counted in the human corpus include 398 couplets, 89 quatrains, and 94 tercets. “Ballads” are not fixed forms but a diverse tradition with internal prosodic differences and national and historical divergences.

I believe that assessing AI classification of “style” requires the thing called “poetic form” because the latter looks like something we can annotate into training data. I do not think this is possible or useful at the chosen scale of genre. Efforts to classify form at scale—whether by humans or human AI trainers—is particularly susceptible to the statistical weakness Nan Z. Da identifies in many computational literary analyses: “To look for homologies in literature, (CLS) must eliminate much of high dimensional data and determine the top drivers of statistically significant variation” [3]. LLMs are, poignantly, entire models of language built off a word-bag theory of language that substitutes the unique dimensions of literary data with orders of magnitude higher vector dimensionality that encodes only semantic value. Rather than follow prior machine learning strategies, we should critique them from a discipline specific understanding of what tokens and embeddings would need to look like. In other words, we should study LLMs as symptomatic of poor theories of poetry and literary language.

5 Prosodic Analysis of LLM Verse

To assess LLM style, I argue, requires the different scale represented by prosody: a scale that is in a sense both much bigger and much smaller. Studying meter is not, primarily, a classification task. Scansion’s goal is not to figure out the meter of a poem or line. Rather, it is to determine the line, poem, poet, and period’s techniques and how they relate to syntax, meaning, and genre functions. Neither of Walsh et al.’s pathbreaking studies define iambic meter beyond “a regular pattern of unstressed and stressed syllables,” although that is sufficient to observe that most GPT poetry is iambic. Walsh et al., suggest about 60%, but Heuser and my own results below suggest it is close

⁴ Of “Poem Written with Buson,” Matthew Rohrer writes “I tried to write lines that sound a lot like them, while using lines of theirs that sound as modern as possible.” <https://www.poetryfoundation.org/poetrymagazine/articles/70164/authors-note-on-writing-with-basho-buson-and-issa>

⁵ On Clifford Geertz’s “blurred genres” and the “fuzziness” of form, see also [1]

⁶ [7]

to 100% unless one specifically prompts for free verse, haiku, and explicitly non-metrical verse (Walsh et al. observe “*GPT-4’s increased ability to not produce iambic meter*”). Meter in these studies remains a proxy for genre or “form”; that may help one determine whether an LLM can escape its doggerel gravity well, but it tells us nothing about an LLM’s verse “style” at the level of the line, phrase, or word. Walsh et al. hint at a deeper level of stylistic detail when they note that “in” and “upon” are very common starting words in GPT-4 poems because they make it easy to write iambic rhythm. But the “iambic” label on its own tells us nothing about how syllable, groups of syllables, word stress, caesurae, intonational stress, and other aspects of prosodic phonology are represented within the latent space of LLMs. These elements are historically generative of English poetry (other languages have other phonologies and optimal meters).

When an LLM manages to produce prosodic structures, it is because those structures became generic enough at some point in digitized literary history to strongly correlate phonological patterns into sequences of words, i.e. concatenations. Shakespearian sonnets are the best example, but a scholar of nineteenth-century American poetry might be unsurprised that Claude excels at the Finnish Kalevala’s trochaic tetrameter or Longfellow’s bestselling 1855 *Song of Hiawatha*. Stylometric analyses of LLM verse permits a new path to AI interpretability. At the same time, LLM’s prosodic dysfluency reminds us of the eccentricity of human language practices described by Roman Jakobson’s “poetic function.”

Here I follow the work of Ryan Heuser, who compares frequency of syllable stress per metrical position in 10-syllable lines in LLMs and sonnets to find “mechanical regularity” that “exceeds even a rewinding of history from free verse to metrical tradition” [8]. Using the scansion tool he developed with colleagues at Stanford, *prosodic.py* [9], I scanned >100k lines of iambic pentameter lines in the xml-formatted Eighteenth Century Poetry Archive (ECPA).⁷

⁷ ECPA is an astonishing achievement of TEI, and I encourage readers to navigate to the site and see the xml encoding for themselves. <https://www.eighteenthcenturypoetry.org/works/bsb18-w0660.shtml>. I am grateful to its Alexander Huber and other developers. For my parsed data and full methodology including set of metrical constraints, see https://github.com/cretic/metricalgpt/saved_corpora_parsers/ECPA_data_and_parsers/ecep_reparsed_full.csv.

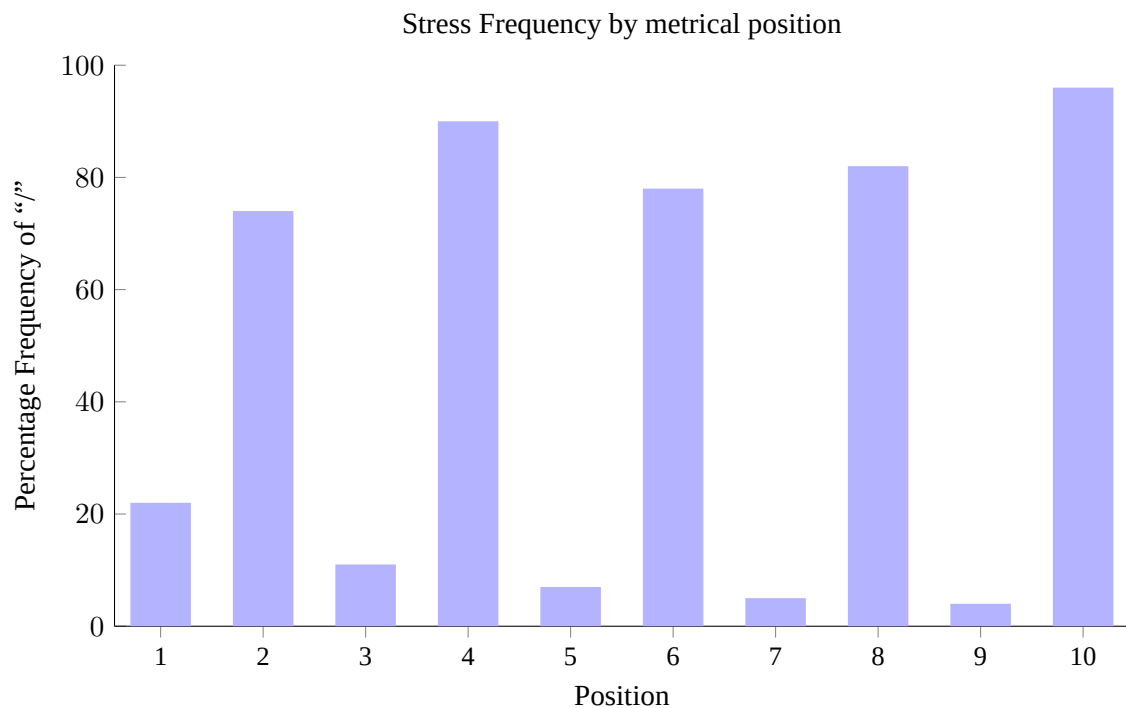


Figure 1: Percentage frequency of / for each metrical position (pentameter)

Compare ChatGPT 4o’s performance when *explicitly* prompted for variations like “trochaic inversion” (Figure 2).

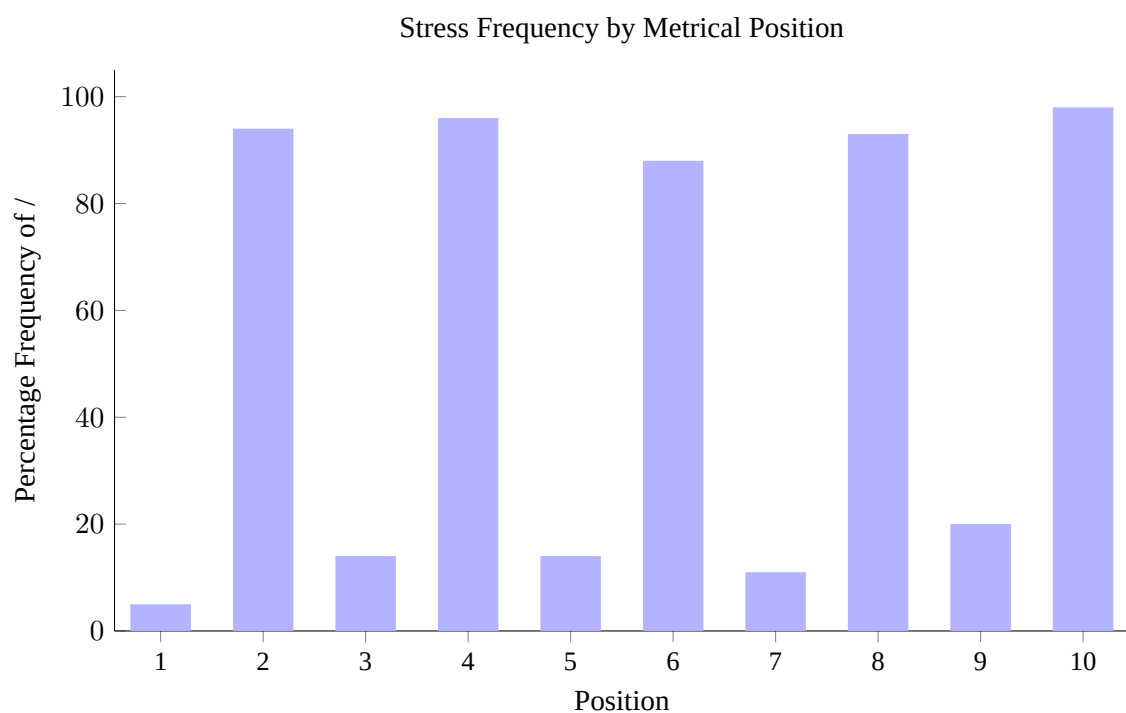


Figure 2: Percentage frequency of / for each metrical position (pentameter)

Such regularity is far in excess of any historical practice, as hand counted by Marina Tarlinskaja, and particularly skewed in the first two positions:

Positions	2	4	6	8	10	Mean 2-8
Pope	78	98	74	86	99	84
Byron	71	80	76	70	94	74
Shakespeare	61	82	73	69	93	71
Frost	61	79	75	74	94	72
Donne	61	80	67	71	78	70
10-syllable prose	38	40	48	38	80	48

Table 2: Tarlinskaja, Stress in Even Positions [12]

A more fine-grained analysis of rhythmical “figures,” i.e. specific stress contours employed historically to draw attention and delight, shows even more mechanicity. For example, the rhythmic figure /xx/, sometimes called a “trochaic inversion” or “inverted foot,” occurs in about 30% of eighteenth-century blank verse lines, usually at the start of the line. Yet prompting for such variation paradoxically decreases frequency of /xx/ (Figure 3).

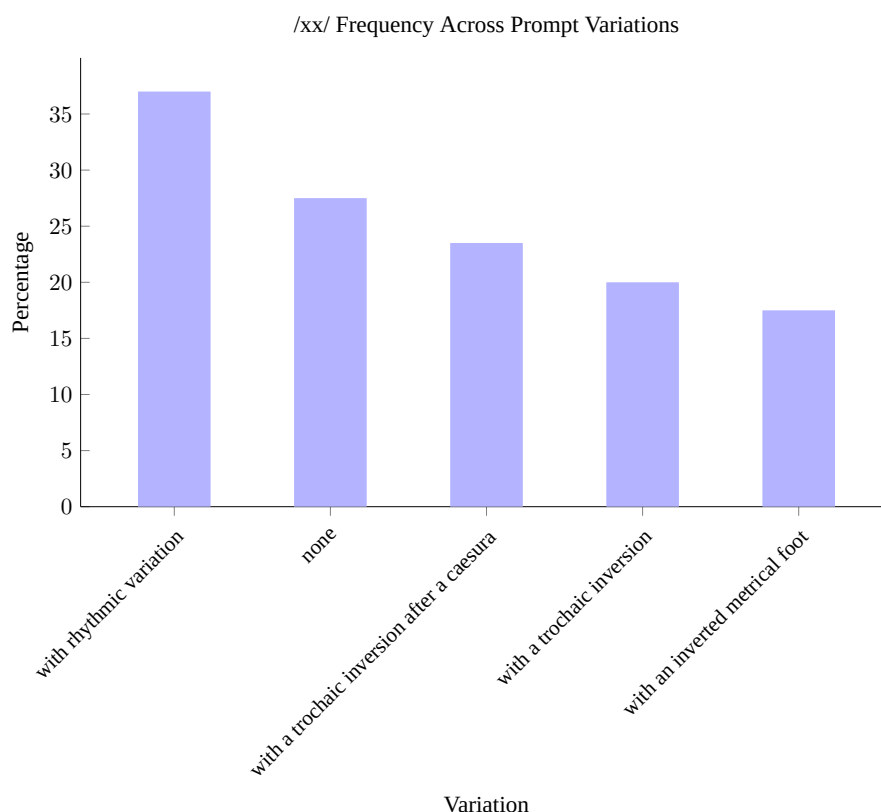


Figure 3: /xx/ frequency in GPT 4o

If we look at the rhythmic figure “xxx,” i.e. three unstressed positions, we see something even more striking: this figure appears in **41%** of ECPA lines, and only 5% of GPT 4o’s blank verse (Figure 4). This shows us that human poets tend towards a lighter line, often one with four stresses total, as a natural consequence of English phonology and the many function words (to, a, in) on which syntax relies after grammatical case was lost. LLMs lack this attention to natural language stress once prompted for poetry and especially when prompted for meter of any sort.

The key finding here is that prompting (and likely any form of instruction tuning) will not produce metrical complexity. The enormous diversity of verse technique compared to LLM’s few species of rigid meter suggests both the obvious absence of phonological awareness and the more interesting presence of prompt frameworks and fine-tunings that pull anything prosody related toward one or two rhythmic patterns.

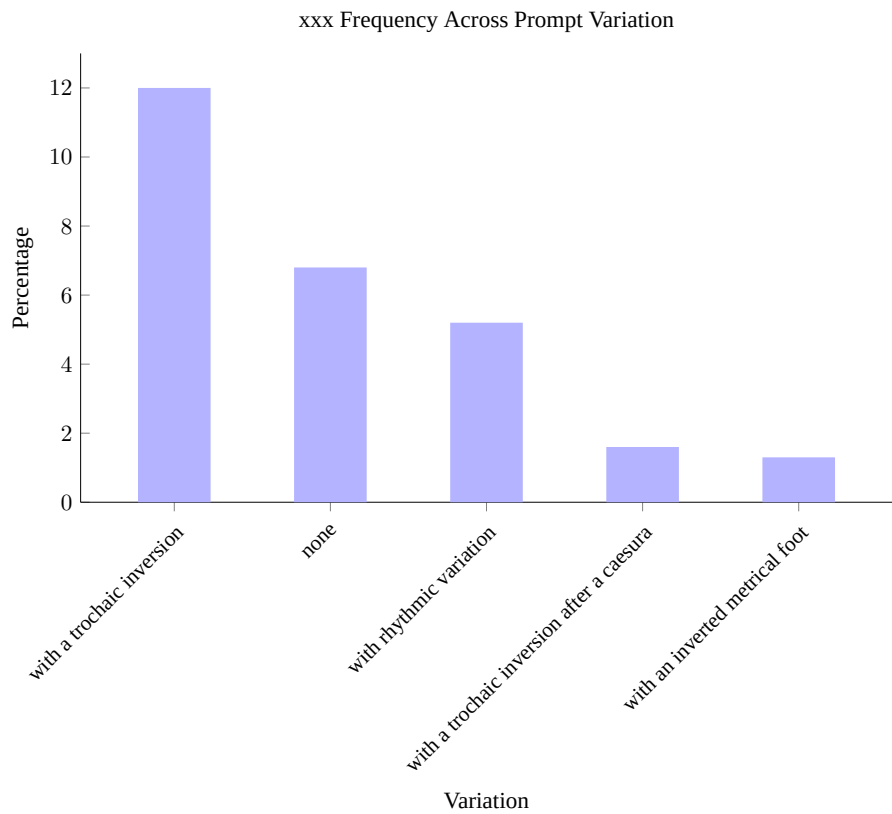


Figure 4: xxx frequency in GPT 4o

6 Training Experiments

To move from simple analysis of decoder models’ output to encoding prosodic features required substantial data collection and curation and symbolic representations of prosody (scansions). First, however I tested what might happen with purely text-based fine-tuning on unusually divergent and complex metrical poetry. Could one push models out of their iambic comfort zone?

6.1 John Milton’s Twin

Compare the following two versions of gemini flash-1.5. On the left, a fine-tuned model trained to generate lines from Paradise Lost. The training data here is 1000 pairings of input=line n, output=line n+1. In other words, if given “Of man’s first disobedience, and the fruit,” gemini tries to produce “of that forbidden tree, who mortal taste.” What metrical system, if any, would be induced? The result was surprising. Next to the fine-tuned model is an un-tuned model given only the system prompt “You are the poet John Milton. When the user inputs a line, you write the next line in Milton’s blank verse.” The output becomes the next input. Early modernists and prosodists may recognize the rhythmic divergence in caesura location (color coded).




PL Fine-tuned (input for each line below and to the right)	Un-tuned “John Milton”
Undubitable, gave me strength, as much 	As Heaven's own grace, to face the coming storm.
As strength the strongest, or to strive, or dare 	To reach the heights where angels fear to tread.
Farthest to place himself beyond the bounds 	Of mortal ken, and dwell in shadows deep.
Of aught, though known. Yet I should still rejoice	To find such solace in the darkest night.
Now to have gained what I have gained, and may	Hereafter win yet greater victory.
Perhaps hereafter gain; but still I curse	The hand that dealt this bitter cup to me.
Her, who with treacherous kiss her son betrayed	And plunged the world in everlasting night.
To slaughter, and most dearly both did pay	The price of treachery with forfeit lives.
For that sad verdict. He that suffers most	Is he who bears the weight of his own sin.
Is he who knows himself nigh finished, and yet	Clings to frail life, though weary of the world.
Feels more for what he once was, and is lost.	Than for the fleeting shadows yet to come.

Figure 5: Gemini flash 1.5, Fine-tuned vs Prompted. Red marks odd-positioned caesurae or strong phrase boundaries; green marks even ones. August 2024

Where Milton performs systematic irregularity, LLMs get stuck dividing their lines after the fourth syllable (with pause or phrasal boundary). The tuned model enjambs, the un-tuned model always endstops. The metrical diversity may be a function of Milton’s syntax, the way he concatenates phrases, and a liberation from the requirement that the line rather than the sentence frame the unit of sense. This was Milton’s central innovation in versification, and here it has the knock-on effect of rhythmical flexibility. Notably, the tuned Milton maintains decasyllabic metricality, suggesting that an LLM may encode these through pure word sequencing constraints.

6.2 metricalBERT

The first challenge of training LLM attention on prosody is the absence of formatted data that includes metrical structure. Inspiration could come from using an analogue of musicBERT to create higher-dimensional tokens including metrical information and phonological representation (IPA or CMUDict encodings). One could otherwise concatenate prosodic data as part of the tokenization process. For example, PoeLM (2022) prepends syllable count and rhyme as control codes.⁸ The creators tag non-verse texts with its syllable count and rhyme to train a model and then control the structure of generated text. Yet only limited efforts have been made to “concatenate tonal information to the character embeddings of an LSTM to create a model that is more phonologically compliant.” No one has done this for accentual-syllabic poetry and English, which present major complications because of complex and hierarchical stress structures.⁹

To compile phonological data for LLM training, I deployed prosodic.py at scale on the ECPA corpus. The result is a >100k line database of scanned iambic pentameter, including a symbolic representation of both stress pattern (rhythm), meter (underlying structure), and a quantitative measure of complexity.¹⁰ Using only the data on metrical complexity, I first fine-tuned several BERT classification models. These indicate a modest potential for inducing metrical awareness into a text-only transformer model. Here is a confusion matrix for “metricalBERT_0_2_4,” trained on the scanned ECPA corpus to predict a line’s metrical complexity as either 0 (simple, no tension), 2, or 4 (a fairly high degree of complexity). Accuracy reached 75%.¹¹

⁸ <https://aclanthology.org/2022.findings-emnlp.268/>

⁹ Another model, poet-vicuna-13B, works by constraining token choice to avoid running over max syllable count (“a logit warper that constrains model vocabulary during the generation process”). See <https://github.com/joehoover/cog-poet-vicuna-13b>. Yet logit warpers modify only the probabilistic process; they are not the same as lexical choice according to qualitative constraints. They show that there is little value in post-embedding layers as “mechanisms for constrained generation” such as forcing syllable counting for “proper metric patterns (e.g. iambic pentameter for our readers who know their scansion).”

¹⁰ https://huggingface.co/datasets/bensglaser/18C_IP_Scanned

¹¹ My training of BERT models was supported by Katelyn DeKeersgieter. The model is available at https://huggingface.co/katelyndekeer/metricalBERT_0_2_4

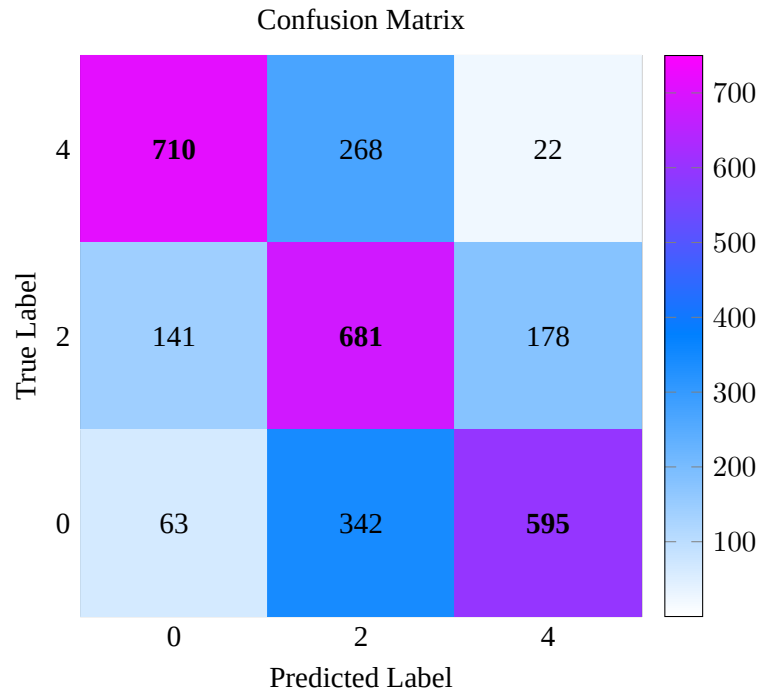


Figure 6: MetricalBERT 0/2/4 Confusion matrix

A purely binary version (0 or 3) approached 80% accuracy.¹² The most promising model, “trochaicBERT”, was trained on the ECPA corpus to differentiate between the 30k lines that have the rhythmic figure “/xx/” (a kind of metrical inversion, sometimes called a trochaic “foot”) and the 70k that do not.¹³ Here precision approached 90-95% (Figure 7).

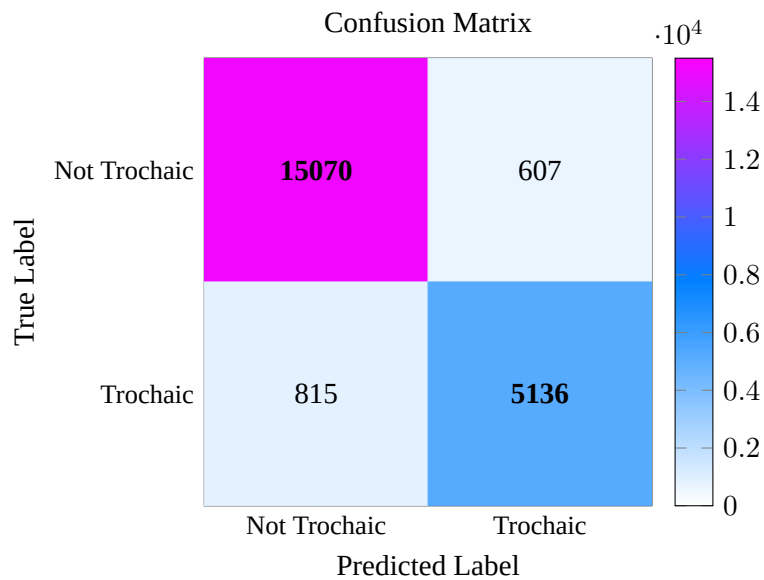


Figure 7: Confusion matrix of trochaicBERT’s classification results.

¹² https://huggingface.co/katelyndekeer/binary_metricalBERT

¹³ https://huggingface.co/bensglaser/trochaic_bert

6.3 metricalT5

metricalBERT showed how classifiers trained on curated prosodic data add phonological information to embeddings of poetic lines. To test the downstream generation and sequence to sequence capacity of transformers, I tagged the dataset to train Google’s T5_large encoder-decoder model using an input-output format. This tracks with the method of training in [13] and [2], relying on annotated datasets converted into sequence transformation tasks in tandem with previously discussed numerical classification of complexity. Prior to this training, I refined our ECPA scansion using openrefine. This removed approximately 2k poorly scanned lines including residual tetrameter and hexameter. To better account for how prosodic.py scores the complexity of lines with uncommon meters, I summed prosodic.py’s score for metrical tension with the Levenshtein distance between the meter prosodic.py assigns and the normative and most common pattern WSWSWSWSWS. Accounting for complexity, which ultimately involves a degree of subjectivity and subtlety, is a work in progress. The more complex, the less rigorous, as perhaps indicated by the confusion matrix in Figure 8.

Input	Target
transform to meter: What fully to describe is past my Art;	WSWSWSWSWS
transform to rhythm: What fully to describe is past my Art;	x/xxx/x/x/
predict complexity: What fully to describe is past my Art;	1
transform to meter: Faint as a chicken’s note that has the pip:	SWWSWSWSWS
transform to rhythm: Faint as a chicken’s note that has the pip:	/xx/x/x/x/
predict complexity: Faint as a chicken’s note that has the pip:	2

Table 3: Sample training data for metricalT5

MetricalT5 achieved high precision (93%) on the “predict complexity” task. It also performs extremely well on sequence transformations, achieving 96% accuracy on both “transform to meter” and “transform to rhythm”.¹⁴ Figure 4 provides two of the 958 rhythm matches and two of the 42 mismatches. MetricalT5’s predictions could facilitate corpus level search and analysis of prosody, for example recurrent rhythmic patterns. It could also guide future LLMs to optimize for more complex and diverse metrical patterns. There are several limitations here, however. Human annotation will always diverge in individual cases; moving outside 18th century poetry and encountering novel words will diminish results; the model has no ability at present to scan non-iambic pentameter lines. It is not, at present, a good tool for the study of individual lines, much less a replacement for teaching and performing (human) scansion.

¹⁴ Performance data is available at https://huggingface.co/bensglaser/metricalT5-large/blob/main/1000line_rhythm_meter_test.csv

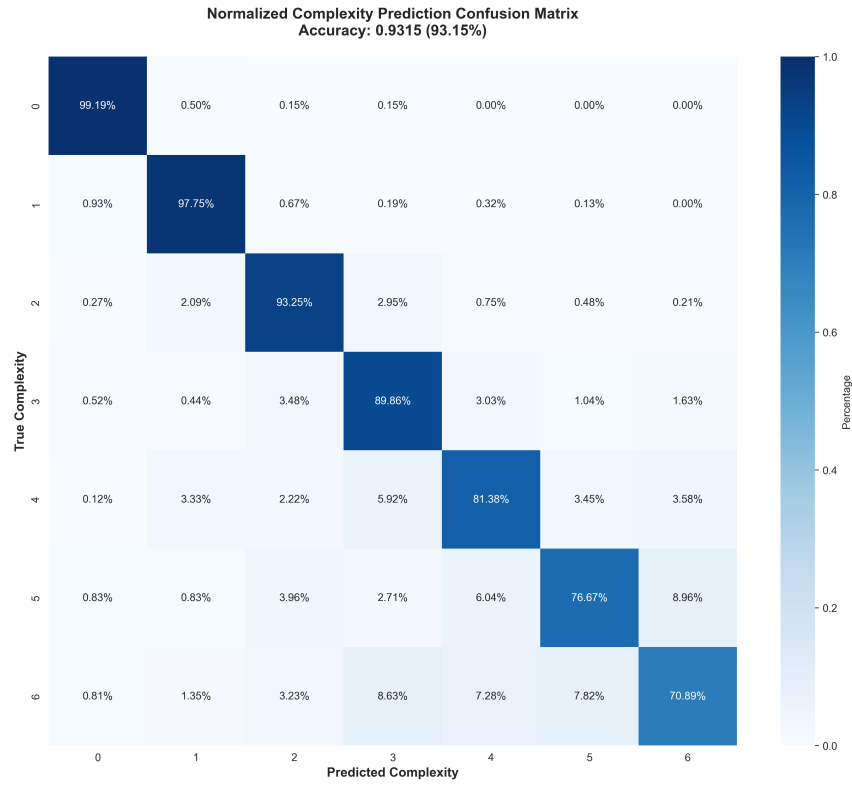


Figure 8: metricalT5 Complexity Confusion Matrix

Input	Target	Predicted	Match
transform to rhythm: Unseen its progress, but its power you find;	x/x//xx/xx/	x/x//xx/xx/	True
transform to rhythm: Is never with impunity defied.	x/xxx/xxx/	x/xxx/xxx/	True
transform to rhythm: Versailles appears – its painted galleries,	x/x/x/x/xx	x/xxx/x/xx	False
transform to rhythm: Let Vales embroidery wear, let Flowers be tinged	//x/xx//x//	//x/xx//xx/	False

Table 4: metricalT5 rhythm transform

Finally, I provide a comparative analysis of Shakespeare’s Sonnet 129 using metricalT5, prosodic.py with human emendations, and a few-shot prompting of Claude’s Sonnet 4.5 model. For Claude, I provided a scansion of Sonnet 116 and the prompt “You are a poetry scholar who scans lines of poetry to determine 1) their underlying metrical pattern, 2) the stress pattern of their performance, and 3) the complexity of the line. Complexity is a function of the distance between (1) and (2).” As with the ECPA training data I regularized spelling. Error distance is measured via Levenshtein distance, namely how many character edits are required to match values. A unified table of these results is available at <https://github.com/cretic/t5-metrical-poetry>.

#	line
1	The expense of spirit in a waste of shame
2	Is lust in action; and till action, lust
3	Is perjured, murderous, bloody, full of blame,
4	Savage, extreme, rude, cruel, not to trust,
5	Enjoyed no sooner but despised straight,
6	Past reason hunted; and, no sooner had
7	Past reason hated as a swallowed bait
8	On purpose laid to make the taker mad;
9	Mad in pursuit and in possession so,
10	Had, having, and in quest to have, extreme;
11	A bliss in proof and proved, a very woe;
12	Before, a joy proposed; behind, a dream.
13	All this the world well knows; yet none knows well
14	To shun the heaven that leads men to this hell.

Table 5: Shakespeare Sonnet 129

#	prosodic complexity	t5 complexity	claude complexity	t5 error	claude error
1	2.0	2.0	1.0	0.0	1.0
2	1.0	1.0	1.0	0.0	0.0
3	0.0	3.0	1.0	3.0	1.0
4	4.0	5.0	3.0	1.0	1.0
5	2.0	3.0	2.0	1.0	0.0
6	3.0	3.0	4.0	0.0	1.0
7	2.0	2.0	3.0	0.0	1.0
8	0.0	0.0	0.0	0.0	0.0
9	2.0	2.0	2.0	0.0	0.0
10	1.0	1.0	4.0	0.0	3.0
11	0.0	0.0	2.0	0.0	2.0
12	0.0	0.0	0.0	0.0	0.0
13	3.0	4.0	1.0	1.0	2.0
14	2.0	2.0	1.0	0.0	1.0
average error				0.429	0.929

Table 6: Complexity Analysis: T5 vs Claude

Line	Prosodic Meter	T5 Meter	T5 Lev Dist	Claude Meter	Claude Lev Dist
1	WWSWSWSWSWS	WWSWSWSWSWS	0.0	WSWSWSWSWS	1.0
2	WSWSWSWSWS	WSWSWSWSWS	0.0	WSWSWSWSWS	0.0
3	WSWSWSWSWS	WSWSWWSWSWS	1.0	WSWSWSWSWS	0.0
4	SWWSWSWSWS	SWWSWSWSWS	0.0	SWSWSWSWWS	2.0
5	WSWSWSWSWS	WSWSWSWSWS	0.0	WSWSWSWSWS	0.0
6	WSWSWSWSWS	WSWSWSWSWS	0.0	SWSWSWSWWS	2.0
7	WSWSWSWSWS	WSWSWSWSWS	0.0	SWSWSWSWWS	2.0
8	WSWSWSWSWS	WSWSWSWSWS	0.0	WSWSWSWSWS	0.0
9	WSWSWSWSWS	WSWSWSWSWS	0.0	SWSWSWSWSW	2.0
10	WSWSWSWSWS	WSWSWSWSWS	0.0	SWSWSWSWSW	2.0
11	WSWSWSWSWS	WSWSWSWSWS	0.0	WSWSWSWSWW	1.0
12	WSWSWSWSWS	WSWSWSWSWS	0.0	WSWSWSWSWS	0.0
13	WSWSWSWSWS	SWWSWSWSWS	2.0	WSWSWSWSWS	0.0
14	WSWSWSWSWWS	WSWSWSWSWS	1.0	WSWSWSWSWS	1.0
average distances			0.286		0.929

Table 7: Meter Analysis: T5 vs Claude

Line	Prosodic Rhythm	T5 Rhythm	T5 Lev Dist	Claude Rhythm	Claude Lev Dist
1	xx/x/x/x/x/	xx/x/x/x/x/	0.0	x/xx/xx/x/	3.0
2	x/x/xxx/x/	x/x/xxx/x/	0.0	x/xx/x/x/x/	2.0
3	x/x/x/x/x/	x/x/xx/x/x/	1.0	x/x/xx/x/x/	1.0
4	/xx//x/x/	/xx//x/x/	0.0	/x/x//x/x/	2.0
5	x/x/xxx/x/	x//xxx//	2.0	x//xx/x/x/	2.0
6	//x/xxx/x/	//x/x//x/	2.0	//x/xx//x/	1.0
7	//x/xxx/x/	//x/xxx/x/	0.0	//x/xx/x/x/	1.0
8	x/x/x/x/x/	x/x/x/x/x/	0.0	x/x/x/x/x/	0.0
9	/xx/x/x/x/	//x/x/x/xx	2.0	/xx/xx/x/x	2.0
10	//xxx/x/x/	x/xxx/x/x/	1.0	//xxx/x//x	2.0
11	x/x/x/x/x/	x/x/x/x/x/	0.0	x/x/x//x/x	2.0
12	x/x/x/x/x/	x/x/x/x/x/	0.0	x/x/x/x/x/	0.0
13	//x//x/x/	/xx//x//	2.0	x/x//x/x/x/	2.0
14	x/x/x//xx/	x/x/x//xx/	0.0	x/x/xx/x/x/	2.0
average distances			0.714		1.571

Table 8: Rhythm Analysis: T5 vs Claude

This comparison reaffirms that frontier models rarely align with computer-aided human scansion save for the simplest lines: those with a standard WWSWSWSWS pattern and x/x/x/x/x/ rhythm. Lines like 4 and 10, which begin with stress (i.e. “trochaic” openings), tend to throw off the model. It does not know that iambic meter stays iambic throughout, restores order as the line (and sonnet) proceeds. I admit that fine-grained analysis of individual lines feels inadvisable for tools aimed at corpus level study. Training metricalT5 on 17th century and other poetry might also improve performance; Claude may perform better with different prompting, though the exploration above suggests it will not.

7 Discussion

Even powerful frontier models like Anthropic’s Sonnet 4.5 lack prosodic information and therefore still rely on concatenated “poetic” phrases, resulting in an extremely limited chunking of language that cannot

be at once poetry and flexible versification. The T5 model trained here has value for future scansion and perhaps pedagogy. It could be looped into an agentic workflow, effectively teaching more powerful models to understand and generate a greater diversity of metrical style. This paper makes no value judgment as to whether that is itself desirable. Training models to detect, analyze, and generate less restricted versification is a quixotic task, but can serve as a powerful test of computational humanities' ability to creatively parse poetry as data. It will help us interpret the further reification of "poetry" and poetic form by powerful models increasingly anointed with intelligence on the basis of their poetic abilities. It will allow collaboration with linguistic and musicological studies of the divergence (or convergence) between AI pattern recognition and our symbolic notation practices, cognitive parsing of prosody, and listening to verse.

References

- [1] Butterfield, Ardis. "Fuzziness and Perceptions of Language in the Middle Ages: Part 1: Explosive Fuzziness: The Duel". In: *Common Knowledge* 18, no. 2 (2012), pp. 255–266.
- [2] Chakrabarty, Tuhin, Padmakumar, Vishakh, and He, He. "Help me write a poem: Instruction Tuning as a Vehicle for Collaborative Poetry Writing". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (2022), pp. 6848–6863.
- [3] Da, Nan Z. "The Computational Case against Computational Literary Studies". In: *Critical Inquiry* 45, no. 3 (Mar. 2019), pp. 601–639.
- [4] Davis, Ernest. "ChatGPT's Poetry is Incompetent and Banal: A Discussion of (Porter and Machery, 2024)". 2024. URL: <https://cs.nyu.edu/~davise/papers/GPT-Poetry.pdf>.
- [5] Elam, Michele. "Poetry Will Not Optimize; or, What Is Literature to AI?" en. In: *American Literature* 95, no. 2 (June 2023), pp. 281–303. ISSN: 0002-9831, 1527-2117. DOI: 10.1215/00029831-10575077. URL: <https://read.dukeupress.edu/american-literature/article/95/2/281/344231/Poetry-Will-Not-Optimize-or-What-Is-Literature-to> (visited on 11/20/2024).
- [6] Glaser, Ben. *Modernism's Metronome : Meter and Twentieth-Century Poetics*. Baltimore: Johns Hopkins University Press, 2020.
- [7] Glaser, Ben. "White Things: Form, Formalization, and the Use of". In: *New Literary History* 54, no. 4 (2024).
- [8] Heuser, Ryan. "Generative Aesthetics: On formal stuckness in AI verse". In: *Journal of Cultural Analytics* 10, no. 3 (Oct. 2025).
- [9] Heuser, Ryan, Falk, Joshua, and Anttila, Arto. "Prosodic (software)". 2010. URL: <https://github.com/quadrismegistus/prosodic>.
- [10] McCoy, R. Thomas, Yao, Shunyu, Friedman, Dan, Hardy, Mathew D., and Griffiths, Thomas L. "Embers of autoregression show how large language models are shaped by the problem they are trained to solve". In: *Proceedings of the National Academy of Sciences* 121, no. 41 (Oct. 2024).
- [11] Porter, Brian and Machery, Edouard. "AI-generated poetry is indistinguishable from human-written poetry and is rated more favorably". In: *Scientific Reports* 14, no. 1 (Nov. 2024), p. 26133.
- [12] Tarlinskaja, Marina. "What is 'Metricality'? English iambic pentameter". In: *Formal Approaches to Poetry: Recent Developments in Metrics*, ed. by B. Elan Dresher and Nila Friedberg. Phonology and Phonetics 11. Berlin: Mouton de Gruyter, 2006, pp. 53–75.
- [13] Walsh, Melanie, Preus, Anna, and Antoniak, Maria. "Sonnet or Not, Bot? Poetry Evaluation for Large Models and Datasets". arXiv:2406.18906. Oct. 2024. DOI: 10.48550/arXiv.2406.18906. URL: <http://arxiv.org/abs/2406.18906> (visited on 10/24/2024).

- [14] Walsh, Melanie, Preus, Anna, and Gronski, Elizabeth. “Does ChatGPT Have a Poetic Style?” Oct. 2024. URL: <http://arxiv.org/abs/2410.15299> (visited on 11/20/2024).