

METRICALARGS: Studying Metrical Poetry with LLMs

Chalamalasetti Kranti¹, Sowmya Vajjala²

¹Department of Linguistics, University of Potsdam, Germany,

²National Research Council, Ottawa, Canada

kranti.chalamalasetti@uni-potsdam.de, sowmya.vajjala@nrc-cnrc.gc.ca

Abstract

Many classical languages have well-studied traditions of poetic meter which enforce constraints on a poem in terms of syllable and phoneme patterns. Such advanced literary forms offer opportunities for probing deeper reasoning and language understanding in Large Language Models (LLMs) and their ability to follow strict pre-requisites and rules in generating text. In this paper, we introduce METRICALARGS, the first taxonomy of poetry-related NLP tasks designed to evaluate LLMs on metrical poetry across four dimensions: **A**nalysis, **R**etrieval, **G**eneration, and **S**upport. We discuss how these tasks relate to existing NLP tasks, addressing questions around datasets and evaluation metrics. Taking the metrical poetry of Telugu language as our example, we illustrate how the taxonomy can be used with LLMs in practice through a quantitative and qualitative evaluation. METRICALARGS highlights the broader possibilities for understanding the capabilities and limitations of today’s LLMs through the lens of metrical poetry. We believe METRICALARGS can also serve as a reference taxonomy for studying and comparing metrical poetry across Indian languages as a starting point, and can be extended to other languages with established metrical poetry traditions.

Keywords: Poetry Analysis, Metrical poetry, Telugu

1. Introduction

There has been a consistent interest in the generation and analysis of the poetic form in NLP and computational creativity research. Although the language of focus is mainly English, there has been some research in a few other relatively high-resource languages such as Chinese. While automated poetry generation/translation has been the most commonly studied problem, there are also several other related topics when we consider more constrained literary-linguistic systems such as meter in poetry, which are governed by their own set of rules and requirements. The language specific nature of these rules, and the customization needed in terms of datasets, evaluation methods etc was a bottleneck in extending such studies in computational creativity into other languages in NLP research.

The advent of LLMs that are capable of in-context learning with very few (or no) examples opens up a possibility of extending this research to other world languages with rich and long standing poetic traditions such as Sanskrit and other Indian languages including Telugu. Many of them involve intricate rules and formal specifications that go beyond free-form generation of poems, which we refer to as metrical poetry throughout this paper (*chandās* in Sanskrit, and similar words in other Indian languages). Existing research on poetry related topics in NLP has largely focused on individual tasks in isolation, with no unified taxonomy to connect them. In this paper, we take the first steps in addressing this gap by first creating a taxonomy of metrical poetry tasks covering four dimensions - Analysis, Retrieval, Generation, and Support, which we call

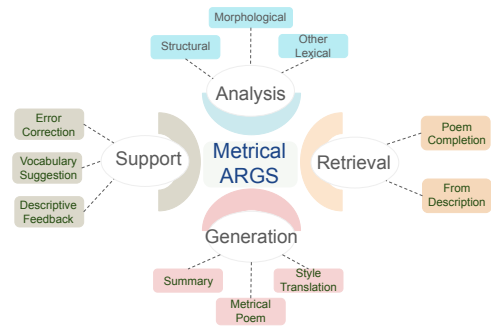


Figure 1: The METRICALARGS taxonomy of tasks for metrical poetry, spanning four dimensions: Analysis, Retrieval, Generation, and Support.

METRICALARGS (Figure 1). We then demonstrate how to use the taxonomy to study the capabilities of LLMs by considering a small test set constructed for Telugu, a Dravidian language with centuries of established metrical poetic tradition (Rao and Shulman, 2020).

From a methodological standpoint, studying metrical poetry with LLMs is important for two main reasons. First, it provides a rigorous testbed for understanding the capabilities of LLMs. Metrical verse requires models to coordinate multiple layers of linguistic competence, including phonology and prosody (to identify and count syllables correctly: a quantitative constraint), morphology and rhythmic structure (to follow metrical and phonological patterns: a structural constraint), and syntax and semantics (to maintain coherence, meaning, and thematic flow: a semantic constraint), all while preserving stylistic and aesthetic consistency. Second, metrical systems are defined by explicit, algorithmic

rules that govern syllable patterns, rhyme positions, and line breaks. This makes them inherently computational in nature and opens up opportunities for developing modeling approaches and evaluation methods that can be integrated with mainstream NLP.

Beyond its methodological significance, studying metrical poetry with LLMs also carries important cultural and pedagogical value. Introduction to the basics of poetic meter happens in the high school level in the standard educational system of many Indian languages, including Telugu, which we use as our test language. Therefore, exploring the relevance of LLMs for metrical poetry also holds a strong pedagogical potential in supporting student learners as well as adult learners. It could also revitalize interest in a classical literary form of the language, assist with cross-linguistic studies of the poetic form, and support other digital humanities research.

With this motivation, we make the following contributions in this paper:

1. We create a taxonomy of tasks around the analysis and generation of metrical poetry and connect them to standard NLP tasks, outlining dataset and evaluation considerations for each task (Section 3).
2. Taking Telugu as the example language, we demonstrate how LLMs can be used for each of these tasks. Our case study (Section 5) serves as an illustrative probe to identify the potential and limitations of using LLMs for metrical poetry related tasks.

To our knowledge, this is the first paper to propose a unified taxonomy of tasks for metrical poetry in NLP, exploring how and where LLMs can support them, rather than focusing on one specific task. Further, this is also the first paper that assesses LLMs on Telugu metrical poetry.

2. Related Work

Most work related to the poetic form in NLP research has focused on poem generation (Ghazvininejad et al., 2016; Gonçalo Oliveira, 2017; Lau et al., 2018; Van de Cruys, 2020; Ormazabal et al., 2022) including recent research involving LLMs (Belouadi and Eger, 2023; Yu et al., 2024; Qu et al., 2025; Koziev and Fenogenova, 2025). In terms of the studied languages, while several world languages such as English (Chakrabarty et al., 2022; Walsh et al., 2024), Chinese (Pan et al., 2023; Ma et al., 2023), and Arabic (ElOraby et al., 2022; Alghallabi et al., 2025) are more widely studied, there is a smaller amount of research on languages such as Portuguese (Valença and Calegario, 2025) and Russian (Koziev and Fenogenova,

2025). Among the Indian languages, previous work focused only on Sanskrit meter (e.g., Jagadeeshan et al., 2026).

Tasks such as poetry analysis (Kao and Jurafsky, 2012; Kesarwani et al., 2017; Gopidi and Alam, 2019; Kurzynski et al., 2024; Sandhan et al., 2025; Jadhav et al., 2025) and translation into a given style/language (Genzel et al., 2010; Ghazvininejad et al., 2018; Chakrabarty et al., 2021; Wang et al., 2024) were also explored in the past, and there is a small amount of research on scansion and metrical analysis (Agirrezabal et al., 2017; Valença and Calegario, 2025; Agirrezabal et al., 2016). There is also some interest in exploring the pedagogical relevance of NLP based tools for poetry generation and analysis (Zhipeng et al., 2019; Rosa et al., 2025), and in the development of tools to scansion a poem i.e., identify the metrical patterns in a poem through rules (Terdalkar and Bhattacharya, 2023). To our knowledge, individual tasks are considered in isolation so far, disconnected from each other, due to a lack of a common taxonomy. Among the Indian languages, only Sanskrit has been studied to some extent across these tasks in the past NLP research (Sandhan et al., 2025; Jadhav et al., 2025; Terdalkar and Bhattacharya, 2023). Despite consistent academic interest in this topic within the NLP community, there has been no categorization of its specific sub-tasks. This paper addresses the issue of building a common taxonomy across related tasks for metrical poetry and introduces a new language, Telugu, into this line of research.

3. Metrical Poetry and NLP Tasks

Meter in poetry can be described as a controlled linguistic system that provides a rhythmic structure to the poems. Meters are typically characterized by rules governing the syllabic and/or sound patterns of the words in a poem, which control the eventual makeup of the poem. We will use syllable based meters as our use case for the rest of this paper. Although many languages of the world have established metrical traditions specific to that language, the typical process of creating a metrically compliant poem across languages consists of similar steps such as: choosing a metrical form, composing lines that fit the meter, abiding by its rhyme and pattern restrictions, and achieving some form of balance between form and meaning. Thus, it may be possible to construct a generalized taxonomy of tasks that can support comparisons across languages considering the common process as the starting point.

With that motivation, we identify a task taxonomy related to understanding and producing metrical poems (Figure 1) and relate these tasks to the standard tasks studied in NLP research (see Figure 2),

as that would enable us to define standard metrics for training and evaluation. We identify four broad categories of tasks: *Analysis, Retrieval, Generation and Support*, and describe them further, looking into potential means of dataset construction and evaluation. Since all of them require deeper understanding and reasoning about the linguistic structure, we do not include reasoning as a distinct task. We take the Telugu meter as a basis for this categorization, but we believe the taxonomy would be adaptable and extendable to other languages.

3.1. Analysis

Analysis tasks concentrate on taking an existing poem and studying its characteristics. We classify analysis into three sub-groups: structural analysis, morphological analysis and other analysis, which are explained below.

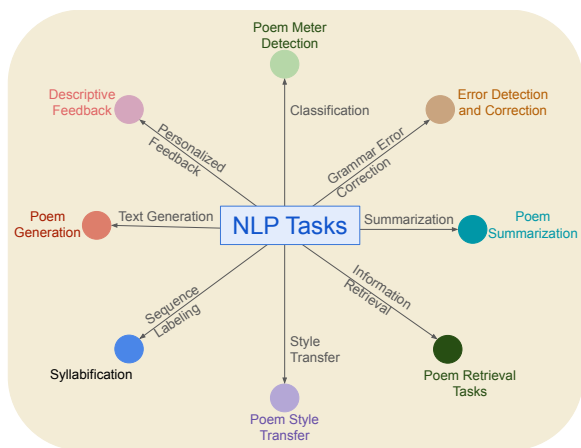


Figure 2: Mapping between METRICALARGS sub-tasks and established NLP tasks. Each poetry-specific task (outer nodes) aligns with a well-known NLP task (center), illustrating how METRICALARGS builds on existing NLP paradigms while extending them to metrical poetry.

Structural Analysis: This refers to the poem’s adherence to a specific metrical form, and can be split into two broad tasks:

1. Syllabification and Syllable Classification:

Given a line from a poem, or the full poem, the task is to identify the correct syllable pattern (or other phonetic length patterns in non-syllabic meters) in the word sequence, and group them together into the appropriate syllable sequences. For example, in Indian languages such as Sanskrit and Telugu, syllable groups are typically 1–4 syllables long, characterized by combinations of heavy (*guru*) and light (*laghu*) syllables, and this task would focus on arranging the given input into such identifiable groups.

2. Mapping a syllable pattern to a meter:

Scansion is the process of identifying the metrical structure of a verse, which involves multiple sub-tasks such as counting the syllables, determining rhyme and other pattern based rules, and verifying them with the rules of the available meters to assign the given pattern to a meter.

The first task can be compared to a standard sequence labeling task such as part-of-speech tagging or named entity recognition in NLP, and the evaluation measures inherited from sequence labeling research can be adapted to this task. Metrical verse identification is a classification problem, as the number of known metrical patterns in a language is typically a fixed number¹. Thus, both the tasks can be measured in terms of accuracy of some form i.e., the number of correctly identified patterns/meters. In terms of data collection, while there are no available datasets of this nature, many classical poems in Telugu and other Indian languages such as Sanskrit are published with an indicator that identifies their meter. Third party rule-based metrical analysis software already exist for some Indian languages such as Telugu², Sanskrit³, which can be used to build a large scale dataset to support a larger evaluation or fine-tuning of the structural analysis abilities of LLMs, or for building verification tools for LLM generated analyses.

Morphological Analysis: This refers to tasks related to glossing of a given poem, which typically involves breaking up the individual words, adding their meanings in a more colloquial language, and tagging the relevant morphological information. In languages with word compounding (many Indian languages, for example), it is a non-trivial process to achieve the appropriate split, and since words can have multiple senses and meanings in context, the task of mapping a word to its right meaning also would involve some form of reasoning. There may also be additional information provided in such glosses, such as person/number/tense information etc. While there are no existing NLP datasets for this task, there are several publicly accessible texts for some languages (e.g., Telugu and Sanskrit) with gloss and plain text summaries for classical metrical poems, which can be utilized to build datasets for this task. There is an already existing body of work on glossing in NLP (Ginn et al., 2023), and evaluation measures from that research can be easily adapted to this task.

¹For Telugu, a listing of 379 meters is at: <https://chandamu.github.io/ChaMdOraajaM.html>

²<https://chandamu.github.io/>

³<https://sanskrit.iitk.ac.in/jnanasangraha/chanda/>

Other Analysis: In digital humanities, as well as in NLP, it is not uncommon to see research studying sentiment, lexical/syntactic/stylistic patterns, authorship attribution and so on, both for prose and verse. Hence, it is natural to study these tasks even with metrical poetry, to understand the capabilities of LLMs in this area. However, while evaluation may be straightforward as such analysis may easily fit into a standard text classification task framework, compilation of relevant datasets for each of these tasks would require substantial human expertise. The role of LLMs in supporting humans in building high quality datasets for this kind of problems should also be explored in this context.

3.2. Retrieval

We refer to tasks related to identifying the right (existing) poems based on user queries as retrieval, which are listed below:

1. Retrieving the poem given its starting words or the first verse or words that appear in the middle or at the end
2. Retrieving the poem from its plain text description, including cross-lingual scenarios (Jagadeeshan et al., 2026).
3. Retrieving the poem(s) that matches in meaning, meter etc.

All of these tasks are similar to search and information retrieval tasks. Considering the large body of classical poetry based literature already available online, collecting datasets at least for the first two of these retrieval tasks should be relatively straightforward. Existing compilations of poem-summary pairs for some languages such as Telugu,⁴ can be utilized as a starting point for creating a larger scale dataset, potentially with synthetically generated paraphrased versions of summaries, which can be useful for evaluation and fine-tuning purposes. In terms of evaluation, the standard retrieval based evaluation measures such as precision/recall/F-score can be used.

3.3. Generation

Generation tasks involve some form of textual generation based on a given description. We identify three generation tasks, described below:

1. **Poem summarization:** Poem to prose translation of a metrical poem, typically written in the classical literary form of the language into plain text. This can be viewed as similar to text summarization which is well-studied in NLP.

⁴<https://huggingface.co/datasets/SuryaKrishna02/aya-telugu-poems>

2. **Poem generation:** generating a novel poem given a textual description and a specified meter. Poem generation under such linguistic constraints was explored in the past in NLP research.
3. **Poem style transfer:** This is a challenging variation to generating a novel poem, where the input is a poem in one meter, and the output is the same content adapted to another meter. This task aligns with other existing research on style transfer in NLP, but in the context of poetry. This kind of metre style transfer is observed in the writings of classical and modern Telugu poets, and hence, can be seen as an advanced text generation task.

For poem to prose summarization, it is easier to create larger scale datasets by tapping into available resources, but for the remaining generative use cases which explore novel and creative content generation, one option is to look at synthetic data generated from LLMs followed by human evaluation. The effectiveness of generation can be evaluated through standard text generation metrics and metrical adherence can be checked through rule based checkers. But, human ratings are a must for other factors such fluency, coherence, adherence to the theme and style, creativity and aesthetics.

3.4. Support

Support tasks explore the role of LLMs in offering support to poets and students learning to write metrical poetry. We identify three main tasks as a starting point.

1. **Error detection and correction:** This refers to the process of identifying metrical, lexical or grammatical errors in the user written poem and offering corrections. This is most similar to the grammatical error detection/correction tasks and word-level translation quality estimation tasks, that have a long history in NLP research.
2. **Vocabulary suggestion:** This task, as the name indicates, offers vocabulary suggestions, but aligning with the metrical constraints of the context. There is perhaps no equivalent existing NLP task as these suggestions need both semantic and metrical compliance.
3. **Descriptive Feedback:** This refers to giving explanation to the user on the text they created and offering suggestions for rewriting. The more recently introduced Grammatical Error Explanation task (Song et al., 2024) is potentially the closest existing NLP task.

All the three support tasks, while being specific to metrical poetry generation, also share some com-

Paper	Lang	Task	METRICALARGS
Kao and Jurafsky (2012)	en	PA	Analysis
Walsh et al. (2024)*	en	PF	Analysis
Valença and Calegario (2025)*	pt	PS	Analysis
Pan et al. (2023)	zh	PA	Analysis
Kurzynski et al. (2024)	zh	PP	Analysis
Jagadeeshan et al. (2026)	sa	PG	Generation
Ghazvininejad et al. (2016)	en	PG	Generation
Belouadi and Eger (2023)*	en	PG	Generation
ElOraby et al. (2022)	ar	PG	Generation
Koziev and Fenogenova (2025)*	ru	PG	Generation
Genzel et al. (2010)	en	ST	Generation
Wang et al. (2024)*	en	ST	Generation

Table 1: Mapping existing metrical poetry works with the proposed METRICALARGS tasks. Lang: Languages supported; PA: Poetry Analysis; PF: Poetic Form; PS: Poetic Scansion; PP: Poem Parallelism; PG: Poem Generation; ST: Style Transfer; * - Indicates the works used LLMs.

monalities with other relevant research on educational applications of NLP, and data creation and evaluation approaches for the related topics can be adapted for these use cases as well. However, it is important to note that they are inherently more challenging tasks than the other ARGS tasks discussed in this paper so far.

From this discussion, it is clear that most of the METRICALARGS tasks can benefit from existing NLP research in related tasks, while introducing challenging new variations. Overall, the METRICALARGS taxonomy demonstrates that there are a wide range of complex tasks related to metrical poetry generation, where LLMs may be relevant and can be further studied. While the taxonomy is created with Telugu meter as its basis (owing to the authors’ familiarity with it), these tasks are not specific to Telugu and can be studied for other Indian languages as the poetic traditions share some commonalities. We expect this task taxonomy to be improved as needed as NLP researchers adapt the taxonomy to poetic traditions in other languages and language families in future.

Figure 2 summarizes the different METRICALARGS tasks and their relation to other standard NLP tasks. Table 1 shows how some of the existing research maps into this taxonomy. Most of the past work appears to have focused on Analysis and Generation tasks, often addressing only a subset of sub-tasks within each category, while Retrieval and Support remain largely unexplored.

4. Applying METRICALARGS for Telugu

We demonstrate the use of METRICALARGS taking Telugu as the test language in this section. Telugu is recognized as one of the classical languages of India (Press Information Bureau, 2024) and has a centuries old literary tradition. Earliest known description of Telugu poetic meters and rules of prosody are from a 6th or 7th century text (Ra-

makrishna et al., 1983, pp.164–165). Telugu poetic meter, while sharing a lot of patterns with Sanskrit poetic tradition has several other native metrical patterns as well. Considering the agglutinative nature of the language, tasks such as breaking up of the syllable sequences into individual words for glossing and summarizing the meaning too offer a range of language processing and reasoning related challenges, along with other tasks around metrical poetry analysis and generation.

To illustrate the use of METRICALARGS taxonomy for Telugu metrical poetry, we curated a dataset of approximately 20 samples for each task (169 samples in total). The intention of using such a small dataset is not to establish a benchmark, but to showcase how current models handle the ARGS tasks and illustrate how to build benchmark datasets using this taxonomy across languages in future. Although modest in size, the dataset covers representative examples of each task. These samples were collected from the official Grade 7–10 Telugu textbooks published by the Andhra Pradesh state government in India. The dataset was prepared and annotated by two native Telugu speakers. Collection of a larger scale dataset covering all the described tasks in the taxonomy is a larger effort beyond the scope of this paper and we hope this pilot study will lead into such benchmarking efforts in future across many languages including Telugu.

Evaluation We considered two proprietary LLMs for output generation: GPT-5 and Gemini-2.5-Pro. For the analysis tasks (syllabification, syllable classification, morphological segmentation, and meter validation) as well as one generation sub-task (summarization), gold references were available. For the remaining categories, such as retrieval, poem generation, and style transfer, multiple valid outputs are possible. In both the cases, we adopted an LLM-as-a-judge approach for evaluation and used Gemini-2.5-Pro as the judge model, supplying it with a gold output for comparison where it is available. The percentage of responses the judge model scores as correct is considered as the measure of performance (with a scale of 0–1, the more the better). All experiments were conducted using the Inspect evaluation framework⁵ in a zero-shot setting, using Telugu prompts, with temperature set to zero. For the subset of cases where there is no single gold standard output, we also conducted a human evaluation in which authors, who are native Telugu speakers reviewed a sample of outputs to verify both the correctness of model predictions and the validity of the LLM-based evaluations⁶.

⁵<https://inspect.aisi.org.uk/>

⁶The dataset and the human evaluation data are both accessible at: <https://huggingface.co/datasets/TeluguLLMResearch/MetricalARGS>

Category	SubCategory	# Q	Accuracy	
			GPT	Gemini
Analysis	SC	20	0.60	0.20
	MA	20	0.20	0.65
	MD	20	0.40	0.50
	FV	6	0.00	0.00
Retrieval	MRV	6	0.00	0.00
	LV	6	0.00	0.00
Generation	PS	20	0.70	0.85

Table 2: Accuracy across different METRICALARGS tasks with gold-output, using LLM-as-a-Judge. SC: Syllabification and Syllable Classification, MA: Morphological Analysis, MD: Meter Detection, FV: Retrieval from First Verse, MRV: Retrieval from Middle/Random Verse, LV: Retrieval from Last Verse, PS: Poem Summarization. #Q indicates the number of questions per task. GPT: GPT-5, Gemini: Gemini-2.5-Pro.

5. Results

We separated the evaluation into two groups depending on whether a gold standard answer was available or not in the dataset. Table 2 presents the results for the tasks where gold references are available. Both the models performed well for the poem summarization task. While the GPT-5 model did better with syllabification, Gemini model did better with morphological analysis. Meter detection was more challenging for both models. The models often misclassified syllable length, leading to downstream errors in meter identification. Metrical rules, such as identifying short and long syllables, are strict and rule-based. LLMs, as next-token predictors, potentially miss fine-grained distinctions that require precise symbolic reasoning, resulting in such errors.

In morphological analysis, both the models generally captured meanings with multiple words instead of single-word glosses, resulting in mismatches with gold annotations, indicating the need for better evaluation measures for that task. Surprisingly, retrieval proved to be a challenging task. Neither model was able to successfully retrieve the complete poem given only the first, last, or a random line as input, resulting in an accuracy of 0 for this task. GPT-5 avoided retrieval altogether by producing follow-up questions (see Figure 3), while Gemini generated paraphrased versions of the poems instead of the actual poems. This could be due to potential coverage issues in the training data, or the models trying to avoid verbatim reproduction of text.

Table 3 reports the results for the tasks where the LLM judge evaluated the generated outputs without a gold answer. GPT-5 appears to be better than Gemini for these tasks, despite the judge being Gemini. Both models achieve near-zero scores

Category	SubCategory	# Q	Accuracy	
			GPT	Gemini
Retrieval	MM	2	0.50	0.00
	PFS	7	0.71	0.29
Generation	RPFW	8	1.00	0.75
	PFP	5	0.40	0.20
	ST	20	0.00	0.00
Support	EDC	10	0.10	0.00
	VS	19	0.47	0.26

Table 3: Accuracy for tasks without gold outputs, evaluated using an LLM-as-a-judge. MM: Retrieval using meaning, PFS: Poem from Summary, RPFW: Riddle Poem from Word, PFP: Poem from a Problem (Samasya in Avadhanam), ST: Style Transfer, EDC: Error Detection and Correction, VS: Vocabulary Suggestion. #Q indicates the number of questions per task. GPT: GPT-5, Gemini: Gemini-2.5-Pro.

for style transfer and error detection and correction. For the generation task in which the models were asked to compose a riddle-style poem, GPT-5 achieved a score of 1.00, while Gemini-2.5-Pro achieved 0.75. In the task of generating a poem from a summary, GPT-5 obtained a score of 0.71 compared to 0.29 for Gemini-2.5-Pro. For style transfer, both models received a score of 0.00. Overall, GPT-5 outputs received higher scores than Gemini-2.5-Pro outputs, even though Gemini-2.5-Pro was used as the judge in both cases. These results suggest that the models are able to produce acceptable outputs for some creative generation tasks, while style transfer remained unsuccessful for both.

Overall, these results provide preliminary insight into the capabilities of LLMs in different METRICALARGS tasks for Telugu, highlighting areas of relative strengths (semantic-level tasks) and weaknesses (retrieval and structural-based tasks). Note that this evaluation covered all the tasks listed in the taxonomy, with two exceptions: "Other Analysis" subset of tasks require expert data curation, and "Descriptive Feedback" is assessed through its components (error correction and vocabulary suggestion). Hence, they are excluded from this case study.

5.1. Qualitative Analysis

To better understand the results, we examine two representative tasks focusing on retrieval and generation respectively. Figure 3 illustrates a retrieval question task where the query provides a first verse and the model is asked to retrieve the complete poem, while the gold standard answer has the expected poem. GPT-5 responded with a follow-up question rather than attempting a retrieval; this behavior was consistent across all samples in this

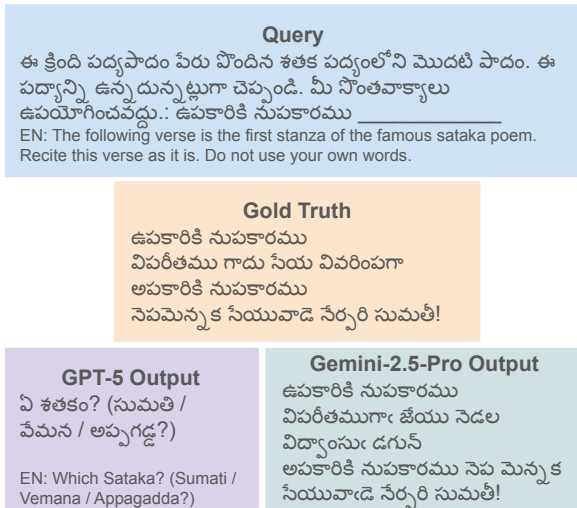


Figure 3: Example from the Retrieval QA task. GPT-5 responded with a clarification question instead of retrieval, while Gemini-2.5-Pro produced text with partial overlap, but did not match the gold reference.

task. Gemini-2.5-Pro produced outputs with some lexical overlap with the gold poem but did not match the target verse, and therefore did not satisfy the exact-retrieval criterion.

Our analysis revealed consistent pattern in how the Gemini-2.5-Pro model used the input verse. When given the first or last verse of the poem, the model retained the input at the appropriate position and attempted to generate surrounding verses. However, the same behavior extended even to the random (middle) verse case, where the input fragment was incorrectly treated as the opening line, and the model generated a continuation around it. This suggests that rather than recalling the original poem, the model interprets the task as one of conditional generation. This task probes the model's ability to match patterns and recall memorized sequences. Given that LLMs generate text through next-token prediction, one might expect them to reproduce a familiar verse when prompted with part of it. However, in our experiments, neither model succeeded, indicating unreliable recall of longer structured texts. Possible reasons include safety guardrails that prevent verbatim reproduction or a limited Telugu training coverage, warranting further tests in other languages.

Figure 4 presents the model outputs and the LLM-judge's reasoning criteria for a poem generation question. In this example, the model is asked to generate a poem in the form of a riddle for a given word. To succeed, the model must first understand the meaning of the word and identify its characteristic features, and then compose a poem that conveys those features indirectly, without explicitly naming the word. This combination of seman-

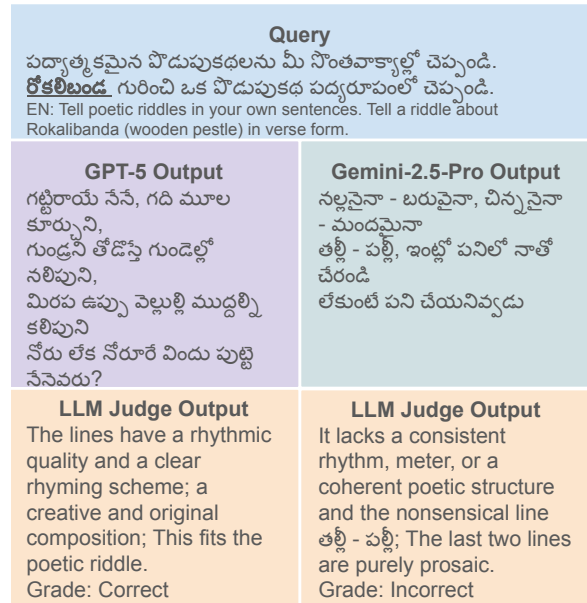


Figure 4: Example from the Riddle-style Poem Generation task. The query asks the model to compose a riddle in verse form about *Rokalibanda* (a wooden pestle).

tic understanding, abstraction, and structured poetic composition makes the task challenging (even for humans). While the LLM-as-a-judge marked the GPT-5 output as correct, human inspection revealed a mismatch: the verse was rhythmically consistent but contained lexical errors and ungrammatical phrasing in Telugu. This highlights both the complexity of the task and a discrepancy between LLM-based and human judgments of fine-grained linguistic accuracy. Taken together, these examples illustrate that while the LLMs can produce outputs that appear fluent or partially aligned with expectations, they often fail to meet stricter criteria such as exact retrieval or grammatical accuracy, underscoring the need for human verification in addition to LLM-based evaluation.

In the Retrieval-Matching Meaning task, lower human scores for GPT-5 stem from its failure to retrieve an existing, popular poem with similar meaning. The model reproduced the input poem verbatim with only the closing verse (last line) changed (see Figure 5), making it a hallucination. The LLM judge, however, marked this as correct, increasing the score relative to human assessments.

In the Generation-Poem from Summarization task, differences between judge and human scores arise from contrasting emphasis on meaning versus form. GPT-5 outputs generally preserve semantic content but often lack poetic coherence or structure, resembling prose or using contextually misplaced words (see Figure 6). The judge, focusing on semantic similarity, overlooks these stylistic deficiencies and assigns higher scores.

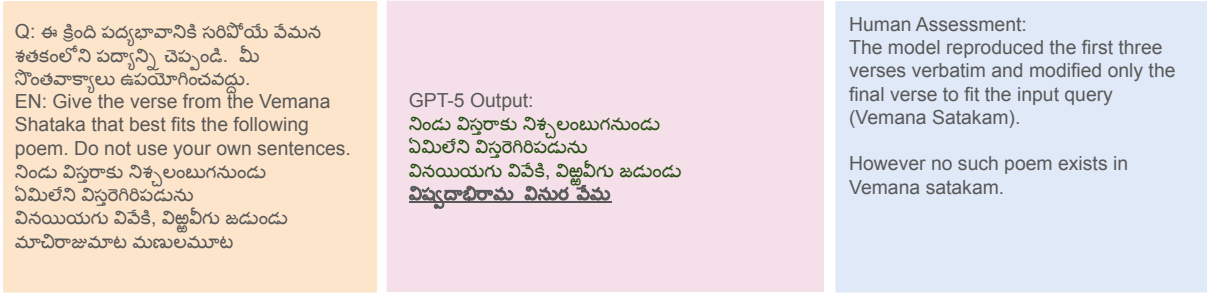


Figure 5: Example of GPT-5 output for the Retrieval–Matching Meaning task, showing the model’s verbatim reproduction of input verses with minor modification.

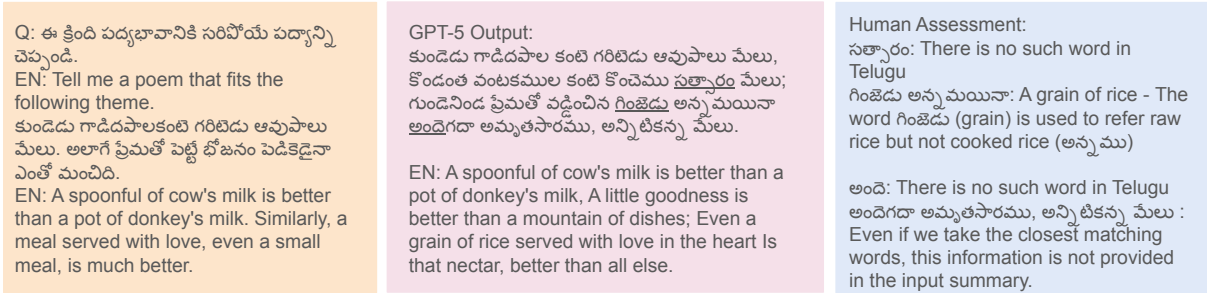


Figure 6: Example of GPT-5 output for the Generation–Poem from Summarization task, showing semantic alignment but lexical and contextual inaccuracies in Telugu usage.

A similar pattern appears in the Generation–Poem (riddle) task, where GPT-5 responses frequently lack logical or poetic structure (see Figure 7) but are still rated correct by the judge. This behavior likely results from the judge evaluating outputs primarily through translation and meaning comparison rather than structural or creative alignment. Consequently, outputs that align in surface meaning but fail in form are treated as correct, while human evaluators apply stricter criteria. In style transfer tasks, model outputs often achieve partial alignment with the target style, sometimes exceeding 70% similarity (see Figure 8).

Overall, these findings suggest that the LLM-as-a-judge primarily emphasizes semantic similarity while neglecting stylistic, structural, and contextual aspects that human evaluators recognize. This tendency results in higher scores for outputs that align in meaning but lack linguistic or creative quality, highlighting the need for evaluation frameworks that account for contextual and stylistic depth.

5.2. Human Evaluation of Model Outputs and Judge Scores

This observation lead us to conduct a human evaluation of LLMs as both generators and judges for these tasks. The authors, both native Telugu speakers, independently evaluated the outputs of both models for the tasks without gold output (i.e., the Tasks in Table 3, which cover about half of the

Category	Sub-Category	JS	A1	A2
GPT-5				
Retrieval	MM	0.50	0.00	0.0
	PFS	0.71	0.29	0.43
	RPFW	1.00	0.00	0.00
Generation	PFP	0.40	0.20	0.20
	ST	0.00	0.00	0.00
Support	EDC	0.10	0.00	0.00
	VS	0.47	0.26	0.26
Gemini-2.5-Pro				
Retrieval	MM	0.00	0.00	0.0
	PFS	0.29	0.14	0.14
	RPFW	0.75	0.63	0.50
Generation	PFP	0.20	0.20	0.20
	ST	0.00	0.00	0.00
Support	EDC	0.00	0.00	0.00
	VS	0.26	0.11	0.16

Table 4: A Comparison of Human and LLM judge evaluations. MM: Retrieval using meaning, PFS: Poem from Summary, RPFW: Riddle Poem from Word, PFP: Poem from a Problem (Samasya in Avadhanam), ST: Style Transfer, EDC: Error Detection and Correction, VS: Vocabulary Suggestion.

dataset) and marked whether each output was correct. These computed scores are then compared with the scores assigned by the LLM-judge in order to assess both the correctness of the model outputs and the reliability of the LLM-as-a-judge framework. Table 4 shows a summary of this comparison.

Overall, the LLM-as-a-judge (JS) reports higher scores than human annotators (A1 and A2) across

fine-tuning datasets across different tasks is an obvious next step to pursue in this direction. Building a larger dataset covering all tasks to support a more comprehensive evaluation is an obvious next step in this direction. Evaluating (and extending) the taxonomy for other Indian languages, and potentially other world languages with similar poetic traditions should be considered in future extensions of this line of research. Consider the ongoing interest in Sanskrit poetry in the NLP community (Jagadeeshan et al., 2026; Terdalkar and Bhattacharya, 2023; Sandhan et al., 2025), establishing a METRICALARGS benchmark for Sanskrit would also be a worthwhile direction to pursue.

Overall, we propose that positioning metrical poetry as a testbed opens up new ways to assess and enhance LLM understanding of form-constrained language, while also supporting application areas such as learning tool development and digital humanities research. We hope that this paper serves as a starting point for further research on exploring the relation between LLMs and metrical poetry, and investigating the role of LLMs in understanding other structured linguistic systems like meter across the world languages.

Limitations

We identify two important limitations to this work:

1. The proposed taxonomy used Telugu metrical poetry tradition as the basis, and the coverage of tasks may not be comprehensive enough to cover the poetic traditions across many languages. Additionally, it is possible to imagine tasks such as metrical poem translation between languages (e.g., *translate a given poem in Telugu meter T-A into Chinese meter C-X*), generating or retrieving a poem given an image (e.g., *generate a poem in Telugu meter T-A, or retrieve a poem in Chinese meter C-X, based on the input image*), which this taxonomy does not currently cover. This can be perceived as a limitation, but we intend this paper to be a starting point to raise further discussions on the topic, and hence, we would expect to see improvements and additions to this taxonomy of tasks in near future.
2. Our empirical study was done based on zero-shot prompting using a small dataset of 169 samples (which do cover most of the tasks in the proposed taxonomy), and two state of the art models, and hence, cannot be considered a large scale, comprehensive evaluation of the abilities of LLMs on METRICALARGS tasks. However, we view this evaluation as an essential starting point to identify requirements for a

larger study. For example, our human evaluation would not have been feasible with a large scale dataset, and it did help us identify the short comings of using an LLM-as-a-judge approach for evaluation for some of the sub-tasks. We hope to conduct a larger scale evaluation in future, by using this smaller study as the basis.

We view these not as limitations to the research topic itself, and hope that this paper laid the necessary foundation to address the above limitations in future.

Acknowledgments

We thank Isar Nejadgholi, Krishnapriya Vishnubhotla and Gabriel Bernier-Colborne for their feedback. Dileep Miriyala’s Chandam (<https://chandamu.github.io/>) inspired us to study metrical poetry with LLMs, and we thank him for creating the tool.

7. Bibliographical References

- Manex Agirrezabal, Iñaki Alegria, and Mans Hulden. 2016. *Machine learning for metrical analysis of English poetry*. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 772–781, Osaka, Japan. The COLING 2016 Organizing Committee.
- Manex Agirrezabal, Iñaki Alegria, and Mans Hulden. 2017. *A comparison of feature-based and neural scansion of poetry*. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 18–23, Varna, Bulgaria. INCOMA Ltd.
- Wafa Alghallabi, Ritesh Thawkar, Sara Ghaboura, Ketan More, Omkar Thawakar, Hisham Cholakkal, Salman Khan, and Rao Muhammad Anwer. 2025. Fann or flop: A multigenre, multi-era benchmark for arabic poetry understanding in llms. *arXiv preprint arXiv:2505.18152*.
- 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.
- Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2022. *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*, pages 6848–6863, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Tuhin Chakrabarty, Arkadiy Saakyan, and Smaranda Muresan. 2021. [Don't go far off: An empirical study on neural poetry translation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7253–7265, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Maryam ElOraby, Mohamed Abdelgaber, Nehal Elkaref, and Mervat Abu-Elkheir. 2022. [Generating classical Arabic poetry using pre-trained models](#). In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 53–62, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Dmitriy Genzel, Jakob Uszkoreit, and Franz Och. 2010. ["poetic" statistical machine translation: Rhyme and meter](#). In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 158–166, Cambridge, MA. Association for Computational Linguistics.
- Marjan Ghazvininejad, Yejin Choi, and Kevin Knight. 2018. [Neural poetry translation](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 67–71, New Orleans, Louisiana. Association for Computational Linguistics.
- Marjan Ghazvininejad, Xing Shi, 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, Austin, Texas. Association for Computational Linguistics.
- Michael Ginn, Sarah Moeller, Alexis Palmer, Anna Stacey, Garrett Nicolai, Mans Hulden, and Mikka Silfverberg. 2023. Findings of the sigmorphon 2023 shared task on interlinear glossing. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 186–201.
- Hugo Gonalo Oliveira. 2017. [A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation](#). In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 11–20, Santiago de Compostela, Spain. Association for Computational Linguistics.
- Amitha Gopidi and Aniket Alam. 2019. [Computational analysis of the historical changes in poetry and prose](#). In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, pages 14–22, Florence, Italy. Association for Computational Linguistics.
- Bhakti Jadhav, Himanshu Dutta, Shruti Kanitkar, Malhar Kulkarni, and Pushpak Bhattacharyya. 2025. [An introduction to computational identification and classification of upamā alaṅkāra](#). In *Computational Sanskrit and Digital Humanities - World Sanskrit Conference 2025*, pages 1–14, Kathmandu, Nepal. Association for Computational Linguistics.
- Manoj Balaji Jagadeeshan, Samarth Bhatia, Pre-tam Ray, Harshul Raj Surana, Akhil Rajeev P, Priya Mishra, Annarao Kulkarni, Ganesh Ramakrishnan, Prathosh Ap, and Pawan Goyal. 2026. [Chandomitra: Towards generating structured Sanskrit poetry from natural language inputs](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 518–534, Rabat, Morocco. Association for Computational Linguistics.
- Justine Kao and Dan Jurafsky. 2012. [A computational analysis of style, affect, and imagery in contemporary poetry](#). In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 8–17, Montréal, Canada. Association for Computational Linguistics.
- Vaibhav Kesarwani, Diana Inkpen, Stan Szpakowicz, and Chris Tanasescu. 2017. [Metaphor detection in a poetry corpus](#). In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 1–9, Vancouver, Canada. Association for Computational Linguistics.
- Ilya Koziev and Alena Fenogenova. 2025. [Generation of Russian poetry of different genres and styles using neural networks with character-level tokenization](#). In *Proceedings of the 9th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2025)*, pages 47–63, Albuquerque, New Mexico. Association for Computational Linguistics.
- Maciej Kurzynski, Xiaotong Xu, and Yu Feng. 2024. [Vector poetics: Parallel couplet detection in classical Chinese poetry](#). In *Proceedings of the 4th International Conference on Natural Language*

- Processing for Digital Humanities*, pages 200–208, Miami, USA. Association for Computational Linguistics.
- Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 2018. [DeepSpear: 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.
- Jingkun Ma, Runzhe Zhan, and Derek F. Wong. 2023. [Yu sheng: Human-in-loop classical Chinese poetry generation system](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 57–66, Dubrovnik, Croatia. Association for Computational Linguistics.
- Aitor Ormazabal, Mikel Artetxe, Manex Agirrezabal, Aitor Soroa, and Eneko Agirre. 2022. [PoeLM: A meter- and rhyme-controllable language model for unsupervised poetry generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3655–3670, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Changzai Pan, Feiyue Li, and Ke Deng. 2023. [TopWORDS-poetry: Simultaneous text segmentation and word discovery for classical Chinese poetry via Bayesian inference](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3372–3386, Singapore. Association for Computational Linguistics.
- Government of India Press Information Bureau. 2024. [Cabinet approves conferring status of classical language to marathi, pali, prakrit, assamese and bengali languages](#). Accessed: 2025-10-08.
- Zhan Qu, Shuzhou Yuan, and Michael Färber. 2025. [Poetone: A framework for constrained generation of structured chinese songci with llms](#).
- Gamapalahalli Ramakrishna, Nagarajarao Gayathri, and Debiprasad Chattopadhyaya. 1983. *An encyclopaedia of South Indian culture*. South Asia Books.
- Velcheru Narayana Rao and David Shulman. 2020. *Classical Telugu Poetry*, volume 13. University of California Press.
- Rudolf Rosa, David Mareček, Tomáš Musil, Michal Chudoba, and Jakub Landsperský. 2025. [EduPo: Progress and challenges of automated analysis and generation of Czech poetry](#). In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 524–542, Albuquerque, USA. Association for Computational Linguistics.
- Jivnesh Sandhan, Amruta Barbadikar, Malay Maity, Pavankumar Satuluri, Tushar Sandhan, Ravi M Gupta, Pawan Goyal, and Laxmidhar Behera. 2025. [Aesthetics of Sanskrit poetry from the perspective of computational linguistics: A case study analysis on śikṣāṣṭaka](#). In *Computational Sanskrit and Digital Humanities - World Sanskrit Conference 2025*, pages 15–36, Kathmandu, Nepal. Association for Computational Linguistics.
- Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2024. [GEE! grammar error explanation with large language models](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 754–781, Mexico City, Mexico. Association for Computational Linguistics.
- Hrishikesh Terdalkar and Arnab Bhattacharya. 2023. [Chandojnanam: A sanskrit meter identification and utilization system](#). In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference*, pages 113–127.
- André Valença and Filipe Calegario. 2025. [Experimenting with large language models for poetic scansion in portuguese: A case study on metric and rhythmic structuring](#). In *Proceedings of the 16th International Conference on Computational Creativity*.
- Tim Van de Cruys. 2020. [Automatic poetry generation from prosaic text](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2471–2480, Online. Association for Computational Linguistics.
- Melanie Walsh, Anna Preus, and Maria Antoniak. 2024. [Sonnet or not, bot? poetry evaluation for large models and datasets](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15568–15603, Miami, Florida, USA. Association for Computational Linguistics.
- Shanshan Wang, Derek Wong, Jingming Yao, and Lidia Chao. 2024. [What is the best way for ChatGPT to translate poetry?](#) In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14025–14043, Bangkok, Thailand. Association for Computational Linguistics.
- Chengyue Yu, Lei Zang, Jiaotuan Wang, Chenyi Zhuang, and Jinjie Gu. 2024. [CharPoet: A Chinese classical poetry generation system based](#)

on token-free LLM. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 315–325, Bangkok, Thailand. Association for Computational Linguistics.

Guo Zhipeng, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. 2019. [Jiuge: A human-machine collaborative Chinese classical poetry generation system](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 25–30, Florence, Italy. Association for Computational Linguistics.