Evaluating Diversity in Automatic Poetry Generation

Anonymous ACL submission

Abstract

Natural Language Generation (NLG), and more generally generative AI, are among the currently most impactful research fields. Creative NLG, such as automatic poetry generation, is a fascinating niche in this area. While most previous research has focused on forms of the Turing test when evaluating automatic poetry generation — can humans distinguish between automatic and human generated poetry — we evaluate the *diversity* of automatically generated poetry, by comparing distributions of generated poetry to distributions of human poetry along structural, lexical, semantic and stylistic dimensions, assessing different model types (word vs. character-level, general purpose LLMs vs. poetry-specific models), including the very recent LLaMA3, and types of fine-tuning (conditioned vs. unconditioned). We find that current automatic poetry systems are considerably underdiverse along multiple dimensions — they often do not rhyme sufficiently, are semantically too uniform and even do not match the length distribution of human poetry. Among all models explored, character-level style-conditioned models perform slightly better. Our identified limitations may serve as the basis for more genuinely creative future poetry generation models.

1 Introduction

A key aspect of creative language generation is the ability to create new, original and interesting text, cf. (Colton et al., 2012; Gatt and Krahmer, 2018; Yi et al., 2020; Elgammal et al., 2017). To date, extremely little attention has been given to the evaluation of originality and creativity in recent creative text generation models such as those for automatic poetry generation, despite renewed interest in the context of recent LLMs (Franceschelli and Musolesi, 2023). In fact, existing automatic poetry generation models are typically not evaluated regarding how different generated poems are from existing poems in the training set but with the *Turing test*: can humans distinguish whether a poem is human authored or automatically generated (Hopkins and Kiela, 2017; Lau et al., 2018; Manjavacas et al., 2019)? However, this form of Turing test and other similar forms of human evaluation may contain an overlooked risk of failure: namely, if the automatically generated instances are (near-)copies of training data instances.

In this work, we fill this gap and evaluate, for the first time, automatic poetry generation systems for their *diversity*. As human evaluation is generally not well suited to assess diversity (Hashimoto et al., 2019), we automatically measure diversity by comparing distributions of generated and existing poems along formal, semantic and stylistic dimensions. This yields much better evidence of the models' creative capabilities in contrast to being mere 'stochastic parrots'.

Our main contributions are: **(i)** we conceptualize diversity of poetry generation systems along different dimensions: diversity on the structural, lexical, semantic and stylistic level; **(ii)** we assess different types of automatic poetry generation systems for diversity: general purpose word and character-level LLMs, both unconditioned and style-conditioned ones, on the one hand, and poetry-specific models, on the other hand; **(iii)** we evaluate each class of model for diversity across the different dimensions, by comparing the distribution of the human authored training data set to the distribution of generated poems. We find that on a distributional level, generated poems are considerably different from human ones. Character-level style-conditioned general-purpose LLMs are most diverse.

Our work prepares the groundwork for truly creative generative AI models (Veale and Pérez y Pérez, 2020) and also has implications for the detection of generative AI (Sadasivan et al., 2023).

We release all code upon acceptance.

2 Related Work

Our work connects to research on diversity and automatic poetry generation, which we now discuss.

Diversity Building systems able to generate diverse output has been a long-standing concern in NLG research (Reiter and Sripada, 2002; van Deemter et al., 2005; Foster and White, 2007) and remains a central issue in neural NLG (Holtzman et al., 2019). The need for careful analysis of NLG systems' diversity – beyond an assessment of the quality or fluency of single-best generation outputs – has been widely acknowledged (Gatt and Krahmer, 2018; Hashimoto et al., 2019; Mahamood and Zembrzuski, 2019; Celikyilmaz et al., 2020; Tevet and Berant, 2021; Schüz et al., 2021). A well-known finding from this line of research is that neural NLG systems typically face a quality-diversity trade-off (Ippolito et al., 2019; Caccia et al., 2020; Wiher et al., 2022): their outputs are either well-formed and fluent or diverse and variable.

Work on evaluating diversity of NLG typically uses automatic metrics that quantify to what extent different outputs by the same system vary (Hashimoto et al., 2019). In practice, though, evaluations of diversity in NLG differ widely across tasks (Tevet and Berant, 2021) and even adopt different notions of diversity (Zarrieß et al., 2021). At the same time, most of these notions focus on lexical or semantic aspects of diversity, e.g., *local lexical diversity*. For instance, Ippolito et al. (2019) compare decoding methods in dialogue generation and image captioning, assessing lexical overlaps in n-best NLG outputs for the same input. Chakrabarty et al. (2022) simply measure the local lexical diversity in automatic generated poems in terms of distinct unigrams. *Global lexical diversity*, on the other hand, measures whether the NLG system generates different outputs for different inputs. For instance, van Miltenburg et al. (2018) define the global diversity of image captioning systems as their ability to generate different captions for a set of inputs, using metrics like the number of types in the output vocabulary, type-token ratio, and the percentage of novel descriptions. Similarly, Hashimoto et al. (2019) view diversity as related to the model's ability to generalize beyond the training set, i.e., generate novel sentences.

Besides lexical diversity, work on open-ended or creative text generation tasks has been interested in diversity at a more general semantic level. For instance, Zhang et al. (2018) and Stasaski and Hearst (2022) aim at building dialogue systems that generate entertaining and semantically diverse responses in chit-chat dialog. Here, semantic diversity has been measured, e.g., with the help of embedding-based similarity (Du and Black, 2019). Chakrabarty et al. (2022) measure creativity of poems via crowd workers: their crowd workers assess which of two poems is more creative.

In our work on diversity in poetry generation, *we complement both lexical and semantic aspects of diversity with aspects of formal diversity. We thus explore whether automatic poetry generation systems are able to capture the 'full bandwidth' of realizations of poetry found in the data distribution with which they have been trained, focusing mostly on global diversity.*

Poetry generation Automatic poetry generation is a long standing dream of AI research, dating back at least to the mid 20th century (e.g., Theo Lutz' *Stochastische Texte*). While early modern systems were heavily hand-engineered (Gervás, 2001), more recent approaches are all trained on collections of human poetry (Lau et al., 2018; Jhamtani et al., 2019; Agarwal and Kann, 2020) but still extensively utilize human guidance e.g. to enforce formal characteristics of poetry such as rhyming (Wöckener et al., 2021). Belouadi and Eger (2023) have recently released a character-level decoder-only LLM (ByGPT5) capable of learning style-constraints such as rhyming without human involvement in model design.

In our work, we explore varying poetry generation models with regard to diversity: poetry-specific models that use hand-engineered architectures as well as general purpose LLMs, including ByGPT5.

3 Diversity in Poetry Generation

We first conceptualize diversity in poetry generation using formal and semantic criteria.

Memorization. In poetry, as in other forms of art, creativity (Sternberg, 1999) plays a central role. A basic aspect of creativity is the models' ability to generate poems that are different from the training data, i.e. have not been memorized as a whole. To examine memorization, we proceed as in Belouadi and Eger (2023). We apply the Ratcliff-Obershelp similarity (Ratcliff et al., 1988) to compare each poem in a sample with poems in the training corpus. If a generated quatrain exhibits a similarity score of $\geq$0.7 with a quatrain in the training data, we classify it as memorized. A quatrain can be divided into

4 verses or 2 couplets; thus, we also inspect memorization at the verse and couplet levels by comparing each verse or couplet in a sample to those in the training data. Higher thresholds for classification are used for these finer-grained comparison levels, as shorter texts have higher chances of being more similar in general. Specifically, a verse with a similarity score $\geq$0.9 or a couplet $\geq$0.8 is considered as memorized. We define the memorization score of a sample as the proportion of memorized quatrains in that sample. How much LLMs memorize from their training data has been a question of central concern recently (McCoy et al., 2023).

Poem length. Within a sample of generated poems, we consider differences at the level of poem length, i.e., their number of tokens, as a basic aspect of diversity at the formal or structural level. We analyze to what extent the length distribution of generated poems differs from the distribution in the training data. We define the length of a quatrain as the number of tokens contained: we eliminate all punctuation symbols and split the remaining text by white space. We report mean length, standard deviation, minimal and maximal length of samples. We additionally deploy distance measures between training data distribution and generated samples, in particular, a metric called histogram intersection (Swain and Ballard, 1991), which measures the intersection area of two normalized histograms (and therefore returns values between 0 and 1).

Rhyme patterns. As a more complex dimension of formal diversity, we consider rhyming as a central aspect that characterizes the structure of a poem. Diversity can then be assessed by comparing rhyme distributions between generated samples and training data. In order to classify rhymes in our samples, we use the same classifier used to annotate QuaTrain (Belouadi and Eger, 2023). We distinguish between true rhymes, which involve different words, and repetitions, which refer to rhymes based on the same word.

Lexical diversity. Lexical diversity is a standard aspect of diversity evaluation in NLG and is used to assess how generation outputs vary in their vocabulary, either at the local text level or at the global corpus level. We use the following metrics to measure the lexical diversity for both the training data and the generated samples: (i) **Averaged type token ratio (ATTR).** We calculate ATTR as the average of all type token ratios (Richards, 1987) (TTRs) for each quatrain in a sample, i.e. as a measure of local lexical diversity. (ii) **Moving average type token ratio (MATTR).** The MATTR (Covington and McFall, 2010) acts on the corpus level and calculates a moving average by sliding through the corpus using a window of fixed size. We deploy this metric as a measure of global lexical diversity. (iii) **Measure of textual, lexical diversity (MTLD).** The MTLD (McCarthy, 2005) is calculated as the average length of a substring that maintains a specified TTR level. MTLD is deployed to measure lexical diversity on a global scale.

	DE		EN	
	QuaTrain	SonNet	QuaTrain	SonNet
Train	253,843	72,526	181,670	51,905
Dev	28,205	8,058	20,186	5,767
Total	282,048	80,584	201,856	57,672

Table 1: Number of quatrains/sonnets in our datasets.

Semantic diversity. Even if a poetry generation system does not directly copy data from the training data, the generated poems may still be semantically very similar to the training data distribution. We employ a multilingual distilled version of Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) as dense vector representations to measure semantic similarity between poems: (i) across the human train set and the generated poems, (ii) within human and generated poems. In particular, for each generated quatrain, we note down the similarity value of the *most similar* human quatrain, then report the average over all those maximum similarity values. We proceed analogously within the human training data and within the automatically generated poems.

4 Experiment Setup

Data We use the QuaTrain dataset published by Belouadi and Eger (2023), which consists of English and German quatrains from different publicly available poetry datasets. The dataset contains human written quatrains but mixes them synthetically: every sequence of four consecutive lines from the underlying human data are included in order to increase dataset size. Besides, it is automatically annotated for meter and rhyme using high-quality classifers (especially for rhyme). Because our focus lies on the diversity of model outputs, we have to avoid repetitions in the training data created by the data augmentation methods used in its creation. To avoid lines appearing multiple times, we

Class	Model	Smaller	Larger	Lang
Poetry-specific	DeepSpeare	-	-	de/en
	SA	-	-	de/en
Unconditioned / Conditioned LLMs	ByGPT5	140m	290m	de/en
	GPT2	117m	774m	de/en
	GPTNeo	125m	1.3b	en
	LLaMA2	7b	13b	de/en
	LLaMA3	8b		de/en

Table 2: Models used in this work. The 'Smaller' and 'Larger' columns display the sizes of the models considered. The 'Lang' column indicates for which languages the models were trained.

first parse the dataset sequentially, eliminating quatrains that overlap the preceding one. Because this method does not eliminate all overlaps, we then use a heuristic, deleting the ten percent of the quatrains which have the biggest overlap with other quatrains until there is no overlap remaining. We refer to the resulting dataset (again) as `QuaTrain`.

`QuaTrain` is split into train and dev sets using a ratio of 9:1; we do not keep a test set since no held-out human data is needed for generation or evaluation. Further, as some models used in this work are designed to process sonnets and/or limerick data, we create pseudo sonnets for them, denoted as `SonNet`. Specifically, for each sonnet, we randomly draw three quatrains and one couplet from the corresponding data split of `QuaTrain`, ensuring that each comes from a different original quatrain. Table 1 provides the data sizes.

Models We use 2 different model classes:

- **Poetry-specific Models**: We select two models that integrate LSTM language models with additional components to generate quatrains with rhymes. ***DeepSpeare*** (Lau et al., 2018) utilizes a pentameter model to learn iambic meter and a rhyme model to distinguish between rhyming and non-rhyming words. ***Structured Adversary*** **(*SA*)** (Jhamtani et al., 2019) learns to rhyme in an adversary setup, where a language model aims to generate poems misclassified by the discriminator, while a discriminator is trained to differentiate between generated and real poems. *Both models can take sonnets as input during training and output quatrains during inference*. For more detailed model descriptions, see Appendix A.1.
- **General Purpose LLMs**: We consider several decoder-only transformer-based models, encompassing both (sub)word- and character-level models, as well as older and very recent models. We choose two model families from the GPT series, GPT2 (Radford et al., 2019) and GPT-Neo (Black et al., 2022) (a replicated version of GPT3 by EleutherAI[1]), two from the LLaMA series, LLaMA2 (Touvron et al., 2023) and LLaMA3 (AI@Meta, 2024), and the *character-level* ByGPT5 (Belouadi and Eger, 2023). Except for LLaMA3, we consider one smaller and one larger variant within each model family based on model size. We train each model in both **unconditioned and conditioned** manners, with rhymes and meters exposed during training in the latter case. For all LLMs, we employ consistent **decoding** strategies for generation: we use the default settings of the LLaMA2 chat models on Hugging Face[2] but limit the number of newly generated tokens to 100 for the word-level models and 300 for the character-level ByGPT5 models.

We end up with a total of 36 models for German and English, categorized into three groups: 1) poetry specific LSTM-based models, 2) unconditioned LLMs, and 3) conditioned LLMs, as summarized in Table 2. `SonNet` is used for training 1), while `QuaTrain` is used for 2) and 3), separately for each language. We train all models using early stopping based on the perplexity/loss observed in the dev sets (details see Appendix A.2), as overfitting may negatively bias certain metrics like memorization rates. To distinguish between the different sizes and training manners of the LLMs, we use the following notation: a subscript of S/L indicates whether it is a smaller/larger version, and a superscript of "con" stands for conditioned training. E.g., $GPT2_S$ and $GPT2_S^{con}$ represent the unconditioned and conditioned trained GPT2 small models, respectively.

5 Evaluation

From each model, we randomly draw 1000 generated poems. Whenever we do a direct comparison between training and generated data (e.g. when comparing lexical diversity), we randomly draw 10 samples of size 1000 (matching the sample size) from the train set and use mean results as representatives. We deploy this strategy to mitigate the large discrepancy in size between human data and generated poems.

[1] https://www.eleuther.ai/

[2] https://huggingface.co/spaces/huggingface-projects/llama-2-7b-chat

	DE		EN	
	verse	couplet	verse	couplet
DeepSpeare	0.83%		0.83%	
SA	**0.40%**		**0.10%**	
ByGPT5$_L$	1.30%*		1.23%*	
ByGPT5$_S$	1.23%		0.93%	
GPT2$_L$	6.85%	0.10%	3.90%	0.10%
GPT2$_S$	8.70%*	0.10%	4.03%*	0.10%
GPTNeo$_L$	-		5.60%*	0.05%
GPTNeo$_S$	-		4.73%	0.10%*
LLaMA2$_L$	4.65%		3.45%*	0.05%*
LLaMA2$_S$	5.45%*		2.48%	
LLaMA3	3.60%		2.88%	0.05%
ByGPT5$_L^{con}$	0.90%*		0.58%	
ByGPT5$_S^{con}$	0.68%		0.75%*	
GPT2$_L^{con}$	4.38%	0.15%*	2.33%*	0.10%*
GPT2$_S^{con}$	6.90%*	0.10%	2.03%	
GPTNeo$_L^{con}$	-		3.88%*	0.05%*
GPTNeo$_S^{con}$	-		3.50%	
LLaMA2$_L^{con}$	4.03%*	0.05%*	2.23%*	
LLaMA2$_S^{con}$	0.70%		0.55%	
LLaMA3^{con}	2.33%		1.65%	

Table 3: Verse- and Couplet-level memorization rates (lower rates are better). Only non-zero entries are displayed. We underline the higher ones between the same models with different training methods, and mark those between the same models of varying sizes with *. The best results in each dimension are bold.

We first investigate structural properties of the generated poems (repetition of instances on a surface level, length distributions, rhyming), then consider lexical and semantic properties.

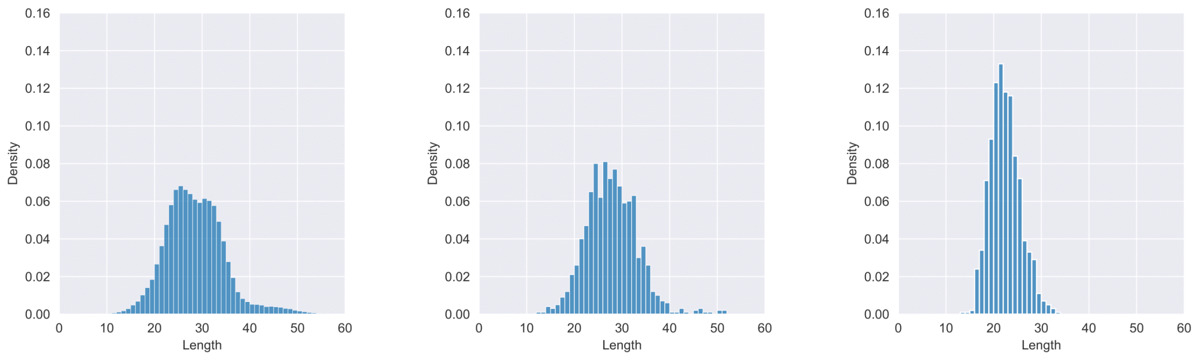
Memorization Table 3 showcases the couplet- and verse level memorization rates. Since all models exhibit zero memorization rates on **quatrain-level**, we omit them in the table.

Considering **couplet-level** memorization, 23 out of 36 models show zero memorization, while 13 models display scores between 0.05% and 0.15%. The poetry-specific models, *SA* and *DeepSpeare*, as well as the character-level ByGPT5 models, exhibit no memorization; in contrast, GPT2 and GPTNeo models show the highest rates on average (up to 0.15% for German and 0.10% for English). When comparing models of the same architecture and training methods but *varying sizes*, differences are found in 6 out of 14 cases. In 5 cases, larger models have 0.05%-0.10% higher absolute memorization scores than their smaller counterparts (the German GPT2^{con} and LLaMA2^{con} models, and the English GPT2^{con}, GPTNeo^{con}, LLaMA2 models); the only exception is the English GPTNeo models, where the smaller one has a 0.05% higher memorization rate. On the other hand, *conditioned models mostly outperform their unconditioned counterparts*: in 4 out of 6 cases where discrepancies in memorization rates exist, the conditioned ones exhibit lower memorization rates, with absolute declines of 0.05%-0.10%.

In the **verse-level** evaluation, the poetry-specific models perform best overall (0.4%-0.83% for German and 0.1%-0.83% for English), followed by the ByGPT5 models (0.68%-1.3% for German and 0.58%-1.23% for English). *SA* is the best individual model, obtaining memorization rates of 0.4% for German and 0.1% for English. Again, GPT2 is worst for German, exhibiting memorization rates of 4.38%-8.7%, whereas, for English, GPTNeo exhibits the highest rates, ranging from 3.5%-5.6%. Concerning different model sizes, we again see that *larger models memorize more than their smaller counterparts*: in 9 out of 14 cases, larger models show higher memorization rates, with an average absolute increase of 0.15%. Here, *each conditioned model exhibits a strictly lower memorization rate compared to its unconditioned counterpart*, with an absolute decrease of 1.47% on average.

In **summary**: (1) No models exhibit severe memorization issues, such as copying entire poems or large portions of poem snippets from the training data. In terms of memorization, (2) among model groups, *the poetry-specific and character-level models are more diverse; SA is the best individual one.* (3) *Larger models are less diverse compared to their smaller versions.* (4) *Conditional training enhances model diversity.*

Length Table 6 (appendix) reports statistics on the length of poems, both human and automatically generated. The mean length of human written poems is 28 in English and 24 in German. Histogram intersection values between samples generated by the models and the human written data range from 0.61 to 0.88 in German (*LLaMA2$_L$* and *SA*) and from 0.48 to 0.92 in English (*GPTNeo$_L$* and *SA*). *While the SA models fit the distribution of the human written poems the best, the character-level ByGPT5 models also perform well consistently with histogram* intersection values between 0.77 and 0.85. The poems generated by German *LLaMA2$_L$* and English *GPTNeo$_L$* are too short and not diverse enough (in terms of standard deviation). The poetry-specific *DeepSpeare* models do not match the human distribution very well either, with intersection values of 0.63 and 0.57 for German and English, respectively. Here, too, poem lengths are too short and not diverse enough. *Con-*



(a) Human (b) SA (c) $GPTNeo_L$

Figure 1: Length distribution of human poems (left), Structured Adversary (middle) and GPTneo-xl (right) for English.

ditioned models seem to fit the training data better across the board, the only exceptions being German $ByGPT5_S$ and English $LLaMA2_S$. Figure 1 illustrates the length distribution of human written poems, *SA* and $GPTNeo_L$ for English.

Rhyme Figures 2 (a) and 3 (a) show the distributions of rhyme schemes in our human training datasets for German and English, respectively. For both languages, less than 15% of all quatrains in training do not rhyme at all (rhyme scheme ABCD). Excluding ABCD, the top 3 dominant rhyme schemes by appearance are ABAB, AABB and ABCB for both datasets, with a total share of approximately 60% in each language. German has a higher proportion of ABAB (above 35%), while English has ABAB and AABB in roughly equal proportions (25%). Table 7 (appendix) reports the entropy of all rhyme distributions and the distance between the human distribution and model distributions, measured in KL divergence. The best, worst and an average model, in terms of KL divergence, are shown in Figures 2 and 3.

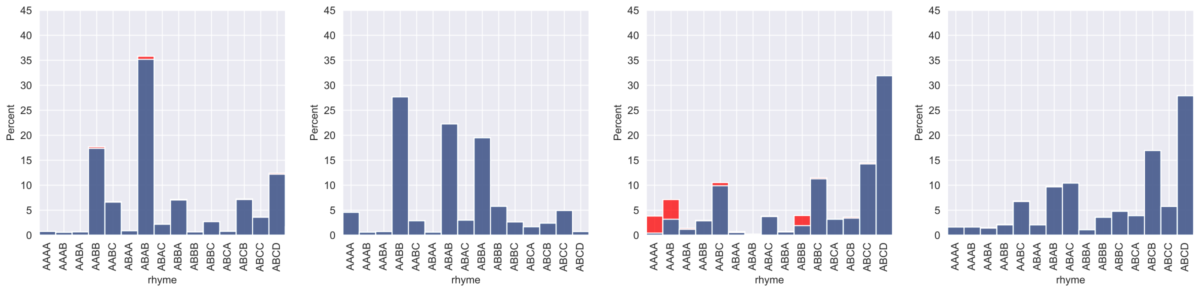
Poetry-specific models: Figure 4 (appendix) shows the distributional plots for *DeepSpeare* and *SA*. We observe that *DeepSpeare* has a very low ratio of ABCD, considerably lower than human poems (less than 5% for both languages). The three dominating patterns are AABB, ABAB, and ABBA which (only) partially agrees with the dominating patterns in the human data. Nonetheless, *DeepSpeare* has the best fit of all models in terms of KL divergence, ranking first for German and second for English. *SA* has a much worse fit and produces considerably too many ABCD patterns (close to or above 30% in both languages). It has one of the worst fits to the human rhyme distributions across all models.

Figures 5 and 6 (appendix) show the distributions of rhyme patterns for **unconditioned LLMs**. Except for *LLaMA3*, all models of this kind have a high distribution of ABCD and consequently a high likelihood of producing non-rhyming poems. Thus, they have the worst fit to the human distribution, on average, among all model classes considered.

Style-conditioned LLMs are shown in Figures 7 and 8 (appendix). In general, this model class matches the human distribution closest in terms of KL divergence. However, no model produces a lot of AABB rhyme pattern which abound in our human training data. Across all models in this class, the fit to the human data is still mediocre at best.

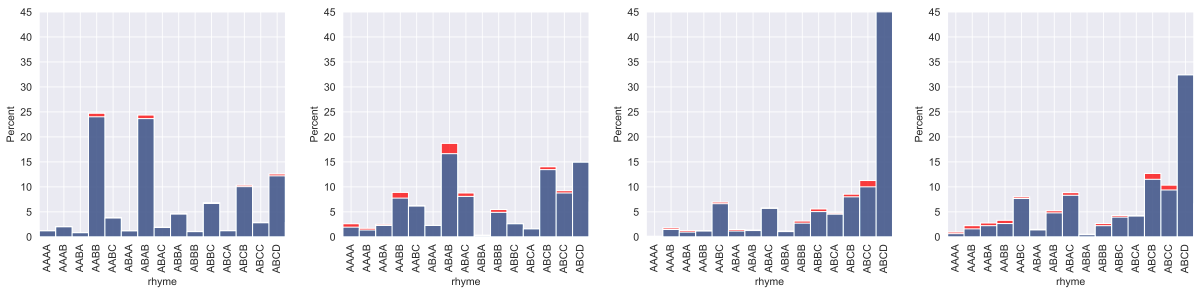
Overall, *most models have clearly higher ABCD rhyming schemes than the human data, thus are underdiverse concerning rhyming. The best model class are style-conditioned LLMs*, however the poetry-specific DeepSpeare model can be considered the best individual model in terms of matching the human rhyme distribution. *The character-level ByGPT5 models perform worse than word-level models without style-conditioning, but with style-conditioning, they outperform the word-level models in terms of match with human rhyme distribution.*

Lexical Diversity. Table 4 shows the lexical diversity results for English and German. *For local diversity (ATTR), most of the models are close to the diversity in human-written poems*, with the traditional models (*DeepSpeare*, *SA*) and the LLaMA exceeding the ATTR values of human-written poems. For German, the least locally diverse poems are generated by $GPT2_S$, in the un/conditioned case, respectively. For English, the least locally diverse models is $GPTNeo_S$, in the un/conditioned case, respectively. The global diversity metrics (MATTR,



(a) **Human** (b) **Best**: *DeepSpeare* (c) **Worst**: *SA* (d) **Avg**: *LLaMA3*

Figure 2: Distribution of rhyme schemes in (a) the human data, and the samples from the (b) best, (c) worst, and (d) average models based on their KL divergence from the human distribution for **German**.



(a) **Human** (b) **Best**: $GPTNeo_L^{con}$ (c) **Worst**: $GPTNeo_L$ (d) **Avg**: $GPT2_S^{con}$

Figure 3: Distribution of rhyme schemes in (a) the human data, and the samples from the (b) best, (c) worst, and (d) average models based on their KL divergence from the human distribution for **English**.

MTLD) show different trends than ATTR, though. The MATTR metric suggests that *most models do not generally achieve the level of diversity found in human poems*: in English, only SA matches and slightly exceeds human diversity, in German, only the $LLaMA2_S^{con}$ and $LLaMA3^{con}$ model exceeds human diversity. According to the MTLD metric, *all models generate severely under-diverse output at the sample level*. Here, the best model in English and German is *SA*, but even *SA* does not come close to the human level of global diversity. According to MTLD, *style-conditioned LLMs consistently outperform their non-conditioned counterparts*, with the English LlaMA2 models being the only exceptions here. Moreover, we observe that model size affects all three lexical diversity metrics, whereby *larger models are more diverse than their smaller counterparts*. The effect of size is most pronounced for GPT2, where ATTR, MATTR and MTLD substantially improve from the small to the larger model variant. It is also noteworthy, though, that the more classical models, *DeepSpeare* and *SA*, generally perform on par with recent transformers and sometimes even outperform them, as in the case of *SA* for global diversity. This shows that unconditional LLMs avoid repetitions at a local level whereas, at the sample level, they generate poems that are lexically much more similar to each other than poems within the human sample. Generally, the MTLD results suggest more pronounced differences between models as well as humans and models than MATTR. This confirms prior studies showing that MTLD does not correlate strongly with TTR-based metrics, capturing different aspects of lexical diversity (McCarthy and Jarvis, 2010).

Semantic Similarity Table 5 presents results for the semantic (cosine) similarity of quatrains: (i) within human and model-generated samples, and (ii) across generated samples and the human data. These results generally confirm the trends for global lexical diversity discussed above. *None of the models generates a sample of poems with a within-sample diversity as low as the human within-sample diversity*. *SA* is the model that achieves the lowest within-sample similarity and the lowest across-sample similarity, suggesting that it deviates most from the patterns in the human training data. Note that *SA* also achieved the best results in global lexical diversity (MATTR and MTLD in Table 4). Moreover, the results on semantic similarity confirm the trends we observed with model size for lexical diversity, but disconfirm the trends for the effect of conditioning. Thus, *we do not see a consistent trend for conditioned models generating samples with lower similarity/higher diversity*.

Model	ATTR (%)	MATTR (%)	MTLD
HUMAN	91.6 / 87.7	90.6 / 87.3	<u>283.1</u> / <u>183.4</u>
DeepSpeare	92.6 / 89.1	87.9 / 84.8	110.0 / 89.7
SA	93.0 / 88.9	**<u>91.0</u>** / 87.8	**215.6 / 162.2**
$ByGPT5_S$	89.7 / 81.5	86.9 / 79.7	135.4 / 66.5
$ByGPT5_L$	91.2 / 82.5	88.1 / 80.5	151.6 / 69.9
$GPT2_S$	86.2 / 79.4	81.2 / 76.4	64.1 / 46.0
$GPT2_L$	94.2 / 87.6	89.5 / 83.5	131.8 / 81.6
$GPTNeo_S$	- / 78.3	- / 74.9	- / 40.1
$GPTNeo_L$	- / 86.8	- / 81.3	- / 61.7
$LLaMA2_S$	92.8 / 89.6	87.7 / 86.8	120.7 / 106.8
$LLaMA2_L$	**<u>94.8</u>** / 90.2	90.2 / 85.7	150.1 / 96.0
LLaMA3	94.4 / **<u>92.7</u>**	89.3 / 87.4	128.0 / 108.1
$ByGPT5_S^{con}$	92.2 / 85.1	89.5 / 83.1	187.1 / 94.6
$ByGPT5_L^{con}$	93.0 / 85.9	90.0 / 83.9	192.6 / 102.5
$GPT2_S^{con}$	89.2 / 84.0	84.2 / 81.9	82.0 / 70.3
$GPT2_L^{con}$	94.2 / 88.0	90.0 / 85.3	137.4 / 90.7
$GPTNeo_S^{con}$	- / 83.1	- / 80.2	- / 61.2
$GPTNeo_L^{con}$	- / 87.0	- / 82.1	- / 69.4
$LLaMA2_S^{con}$	91.1 / 90.0	86.8 / 88.2	104.4 / 109.3
$LLaMA2_L^{con}$	91.9 / 90.8	86.5 / 87.2	100.2 / 101.0
$LLaMA3^{con}$	93.5 / 91.7	89.1 / **<u>88.3</u>**	128.5 / 116.3

Table 4: Lexical diversity metrics for German (first entry) and English (second entry) models. Best results in each dimension are underlined; best among models are in bold.

Model	Within (%)	Across (%)
HUMAN	**55.0 / 48.2**	-
DeepSpeare	59.5 / 52.2	67.8 / 60.8
SA	55.8 / 49.6	**65.9 / 59.4**
$ByGPT5_S$	58.4 / 53.2	68.1 / 61.5
$ByGPT5_L$	58.2 / 52.7	67.9 / 61.6
$GPT2_S$	64.5 / 59.5	69.3 / 63.9
$GPT2_L$	63.6 / 57.6	70.1 / 63.3
$GPTNeo_S$	- / 62.2	- / 63.8
$GPTNeo_L$	- / 60.9	- / 63.9
$LLaMA2_S$	61.0 / 59.4	68.5 / 64.2
$LLaMA2_L$	62.3 / 58.0	68.9 / 62.9
LLaMA3	61.2 / 58.4	69.1 / 63.8
$ByGPT5_S^{con}$	58.4 / 52.2	67.7 / 60.8
$ByGPT5_L^{con}$	57.9 / 50.9	67.6 / 60.3
$GPT2_S^{con}$	64.3 / 59.2	70.1 / 64.3
$GPT2_L^{con}$	62.6 / 57.4	69.7 / 63.1
$GPTNeo_S^{con}$	- / 58.9	- / 64.0
$GPTNeo_L^{con}$	- / 60.3	- / 62.9
$LLaMA2_S^{con}$	66.9 / 57.3	69.3 / 64.0
$LLaMA2_L^{con}$	63.3 / 58.5	69.5 / 62.9
$LLaMA3^{con}$	59.6 / 58.2	68 / 62.3

Table 5: Average maximum semantic similarity values for German (first entry) and English (second entry): (i) within models including the training data (left) and (ii) across models and humans (middle). We bold the best result in each dimension (Lower similarity means higher/better diversity).

For model size, on the other hand, we observe *a general trend towards larger models outperforming their smaller counterparts.*

Which is the most diverse model? We have seen that unconditioned LLMs exhibit poor results across various dimensions of diversity: they often do not rhyme, are lexically underdiverse and do not show sufficient semantic variation. However, character-level models are more diverse than word level models. Style-conditioned models perform better regarding memorization, rhyming, and lexical variation, while deviating less from human poems according to the distribution match of length and rhymes. On the other hand, larger LLMs often outperform their smaller counterparts in semantic and lexical diversity, but they also tend to memorize more from the training data. Character-level style-conditioned LLMs produce overall best diversity results and do not deteriorate as a function of model/training data size. In Appendix A.3, we calculate the average ranks of the models across all 5 dimensions, finding that indeed, for both languages, the conditioned trained ByGPT5 models perform overall best among all models, ranking as the first and second places for German and the first and third places for English. In terms of diversity, poetry-specific *SA* and *DeepSpeare* overall lag only slightly behind character-level LLMs but require more modeling effort from human experts (e.g., in developing rhyming components). The largest word-level LLMs explored in this work, LLaMA2 and LLaMA3, generally perform best among the word-level models; however, they do not exhibit superiority over the style-conditioned character-level models and poetry-specific models as well.

6 Conclusion

Our work is the first and most comprehensive automatic evaluation of poetry diversity, yielding several interesting observations. It shows that an automatic assessment of the diversity of generated poems covers an important blind spot of existing studies. Our evaluations shed light on the fact that none of the state-of-the-art poetry generators is able to match the level of diversity in human poems. Our study also adds a new dimensions to previous work on diversity, by showing that diversity on the level of rhyming is particularly hard to achieve for neural generators and interacts with other dimensions of diversity in poetry generation, i.e., style conditioned LLMs do not only achieve a better match with human rhyme and length distributions, but also higher lexical diversity and lower memorization degree.

7 Limitations

Our work evaluates a range of existing state-of-the-art approaches, such as poetry-specific models like Deepspeare or pretrained LLMs. These models differ in various ways, with respect to their architecture, training scheme, pretraining, and the type of data they expect during training and/or finetuning. In light of these differences, it is difficult to isolate exactly how different aspects of a poetry generator impact on the diversity of its outputs. While our work investigated the influence of the model architecture on a high level (character vs. word), further aspects — and in particular pre-training — may be worth investigating in future work.

Due to the hardware constraints and time limitations, we did not run experiments multiple times to take the averages or optimize the training hyperparameters, which may have introduced a degree of randomness in our results. Indeed, sometimes there have been models behaving inconsistently with others. We expect that a more rigorous training process could increase the consistency in model behaviors and thereby enhance the robustness of our findings. In our initial experiments, we trained GPT2 models with a slightly different setting. Compared to the GPT2 models we mainly reported, these models behave slightly differently. E.g., they exhibit better lexical diversity, as shown by an increase in ATTR from 0.87 to 0.89, MATTR from 0.84 to 0.86, and MTLD from 88 to 101 on average. Similarly, they are also more diverse according to the semantic similarity metrics, which are on average $\sim$0.02-0.03 lower. In contrast, these models perform worse in rhyming; they have a $\sim$10% lower chance of producing rhymed quatrains, and their rhyme distributions are more distant from human distributions (0.27 higher KL divergence). Despite these differences, our findings are generally robust. For instance, conditioned LLMs are still more diverse than their unconditioned counterparts, and larger LLMs are more diverse than their smaller versions, concerning lexical diversity.

8 Ethics Statement

Often, the discussion of creative AI systems in public discourse is surrounded by misconceptions, hypes and even myths (Veale, 2012). Our work contributes to a careful operationalization and objective assessment of the creative capbalities of AI systems in the area of poetry generation.

All the datasets, models and code used in this work are publicly available or will be made available upon publication. We have not collected private or sensitive data and have only used language models with free access, such that our experiments can be fully replicated by anyone.

Generally, our work is concerned with the evaluation of NLG systems; evaluation methods and evaluation metrics (Zhao et al., 2019; Zhang et al., 2020; Yuan et al., 2021; Chen and Eger, 2023; Peyrard et al., 2021) are a well-known and notorious issue in this research field. While a lot of recent work has aimed at improving common practices in human evaluation (Belz et al., 2023) or advancing the study of metrics for quality or fluency of NLG outputs, the evaluation of diversity is comparatively under-researched. In this work, we aimed at providing a range of metrics assessing different aspects of diversity, but could not cover all potentially interesting ways of measuring diversity. Here, future work could look at further aspects of formal and structural diversity (e.g. at the level of syntax, or meter), or other aspects of semantic diversity (e.g. topical diversity, rhetorical figures). Future work could also consider more (diverse) languages and other genres and datasets for poetry.

References

Rajat Agarwal and Katharina Kann. 2020. Acrostic poem generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1230–1240, Online. Association for Computational Linguistics.

AI@Meta. 2024. Llama 3 model card.

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.

Anya Belz, Craig Thomson, and Ehud Reiter. 2023. Missing information, unresponsive authors, experimental flaws: The impossibility of assessing the reproducibility of previous human evaluations in NLP. In *The Fourth Workshop on Insights from Negative Results in NLP*, pages 1–10, Dubrovnik, Croatia. Association for Computational Linguistics.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and

Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2020. Language gans falling short. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

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.

Yanran Chen and Steffen Eger. 2023. MENLI: Robust Evaluation Metrics from Natural Language Inference. *Transactions of the Association for Computational Linguistics*, 11:804–825.

Simon Colton, Geraint A Wiggins, et al. 2012. Computational creativity: The final frontier? In *Ecai*, volume 12, pages 21–26. Montpelier.

Michael A Covington and Joe D McFall. 2010. Cutting the gordian knot: The moving-average type–token ratio (mattr). *Journal of quantitative linguistics*, 17(2):94–100.

Wenchao Du and Alan W Black. 2019. Boosting dialog response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 38–43, Florence, Italy. Association for Computational Linguistics.

Ahmed M. Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. 2017. CAN: creative adversarial networks, generating "art" by learning about styles and deviating from style norms. In *Proceedings of the Eighth International Conference on Computational Creativity, ICCC 2017, Atlanta, Georgia, USA, June 19-23, 2017*, pages 96–103. Association for Computational Creativity (ACC).

Mary Ellen Foster and Michael White. 2007. Avoiding repetition in generated text. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pages 33–40, Saarbrücken, Germany. DFKI GmbH.

Giorgio Franceschelli and Mirco Musolesi. 2023. On the creativity of large language models. *arXiv preprint arXiv:2304.00008*.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Pablo Gervás. 2001. An expert system for the composition of formal spanish poetry. *Knowledge-Based Systems*, 14(3-4):181–188.

Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1689–1701, Minneapolis, Minnesota. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Jack Hopkins and Douwe Kiela. 2017. Automatically generating rhythmic verse with neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 168–178, Vancouver, Canada. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3752–3762, Florence, Italy. Association for Computational Linguistics.

Harsh Jhamtani, Sanket Vaibhav Mehta, Jaime G Carbonell, and Taylor Berg-Kirkpatrick. 2019. Learning rhyming constraints using structured adversaries. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6025–6031.

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.

Saad Mahamood and Maciej Zembrzuski. 2019. Hotel scribe: Generating high variation hotel descriptions. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 391–396, Tokyo, Japan. Association for Computational Linguistics.

Enrique Manjavacas, Mike Kestemont, and Folgert Karsdorp. 2019. A robot's street credibility: Modeling authenticity judgments for artificially generated hip-hop lyrics.

Philip M McCarthy. 2005. *An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (MTLD)*. Ph.D. thesis, The University of Memphis.

Philip M. McCarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42:381–392.

R. Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2023. How much do language models copy from their training data? evaluating linguistic novelty in text generation using RAVEN. *Transactions of the Association for Computational Linguistics*, 11:652–670.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Maxime Peyrard, Wei Zhao, Steffen Eger, and Robert West. 2021. Better than average: Paired evaluation of NLP systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2301–2315, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners.

John W Ratcliff, David Metzener, et al. 1988. Pattern matching: The gestalt approach. *Dr. Dobb's Journal*, 13(7):46.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ehud Reiter and Somayajulu Sripada. 2002. Squibs and discussions: Human variation and lexical choice. *Computational Linguistics*, 28(4):545–553.

Brian Richards. 1987. Type/token ratios: What do they really tell us? *Journal of child language*, 14(2):201–209.

Vinu Sankar Sadasivan, Aounon Kumar, S. Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *ArXiv*, abs/2303.11156.

Simeon Schüz, Ting Han, and Sina Zarrieß. 2021. Diversity as a by-product: Goal-oriented language generation leads to linguistic variation. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 411–422, Singapore and Online. Association for Computational Linguistics.

Katherine Stasaski and Marti Hearst. 2022. Semantic diversity in dialogue with natural language inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 85–98, Seattle, United States. Association for Computational Linguistics.

Robert J Sternberg. 1999. *Handbook of creativity*. Cambridge University Press.

Michael J Swain and Dana H Ballard. 1991. Color indexing. *International journal of computer vision*, 7(1):11–32.

Guy Tevet and Jonathan Berant. 2021. Evaluating the evaluation of diversity in natural language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 326–346, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Kees van Deemter, Emiel Krahmer, and Mariët Theune. 2005. Squibs and discussions: Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24.

Emiel van Miltenburg, Desmond Elliott, and Piek Vossen. 2018. Measuring the diversity of automatic image descriptions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1730–1741, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tony Veale. 2012. *Exploding the creativity myth: The computational foundations of linguistic creativity*. A&C Black.

Tony Veale and Rafael Pérez y Pérez. 2020. Leaps and bounds: An introduction to the field of computational creativity. *New Generation Computing*, 38:551–563.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

Jörg Wöckener, Thomas Haider, Tristan Miller, The-Khang Nguyen, Thanh Tung Linh Nguyen, Minh Vu Pham, Jonas Belouadi, and Steffen Eger. 2021. End-to-end style-conditioned poetry generation: What does it take to learn from examples alone? In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 57–66, Punta Cana, Dominican Republic (online). Association for Computational Linguistics.

Xiaoyuan Yi, Ruoyu Li, Cheng Yang, Wenhao Li, and Maosong Sun. 2020. Mixpoet: Diverse poetry generation via learning controllable mixed latent space. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9450–9457.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Sina Zarrieß, Hendrik Buschmeier, Ting Han, and Simeon Schüz. 2021. Decoding, fast and slow: A case study on balancing trade-offs in incremental, character-level pragmatic reasoning. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 371–376, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 1815–1825, Red Hook, NY, USA. Curran Associates Inc.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 *DeepSpeare* and *SA*

Deepspeare (Lau et al., 2018) is specifically designed for poetry generation. Its core architecture consists of an LSTM language model, a pentameter model (specifically designed to learn iambic meter) and a rhyme model. During training, it takes sonnets as input data (three quatrains followed by a couplet) but ultimately processes the contained quatrains by splitting any given sonnet. The rhyme model processes ending words of quatrain verses and uses a margin-based loss to discriminate between rhyming and non-rhyming words. It is not limited to specific rhyme patterns but assumes that rhymes exist in the data. At inference time, Deepspeare generates quatrains.

Structured Adversary. Like Deepspeare, Structured Adversary (SA) (Jhamtani et al., 2019) incorporates different components: an LSTM language model and a discriminator used to decide whether line endings are typical for poetry. Both components are organized in an adversarial setup, where the language model acts as a generator, trying to generate poems that are misclassified by the discriminator, while the discriminator is trained to distinguish generated poems from real ones. SA is trained with sonnets as input data. At inference time, it generates quatrains.

A.2 Training

DeepSpeare *DeepSpeare* (Lau et al., 2018) leverages pretrained static word vectors. We use `QuaTrain` and `SonNet` to train our own Word2vec embeddings (Mikolov et al., 2013) and the final sonnet models respectively. For the sonnet model training, we use a batch size of 128 and apply early stopping with a patience of 5 epochs; default settings are maintained for the other hyperparameters.

SA We use the same word vectors and training data splits as for *DeepSpeare*. Training *SA* involves 1) pretraining the discriminator's encoder using a publicly available pronouncing dictionary ; 2) training the LM component; 3) training a final aggregated model in a generative adversarial setup. We train the discriminators with a batch size of 128, the LMs with a batch size of 64, and the final sonnet models with a batch size of 128; here, we also implement early stopping with a patience of 5 epochs.

Style-un/conditioned LLMs We train all LLMs on our train set using the paged AdamW optimizer with weight decay of 0.001, a learning rate 4e-05 for 50 epochs, and a cosine learning rate decay with a 3% warmup ratio, early stopping with a patience of 5 epochs. As we run experiments on GPUs with varying memory capacities ranging from 12GB to 80GB, and with models that drastically differ in size. To achieve as much consistency as possible, we either train models with a batch size of 128 or accumulate the batches to reach a size of 128. For LLaMA, we use 4-bit quantization and LORA (Hu et al., 2021); the corresponding parameters are list below:

- target modules: q_proj, v_proj, k_proj, o_proj, embedded_tokens
- lora alpha: 16
- lora dropout: 0.05
- r: 16

A.3 Evaluation Results

Length Table 6 displays the length related statistics.

Rhyme Table 7 shows the entropy of the rhyme distributions in each sample as well as the distances of the distributions to that in the human data, measured by KL divergence. Figure 3 demonstrates the human rhyme distribution as well as the best, worst, and an average fit distributions in terms of KL divergence. Figures 4, 5/6, and 7/8 demonstrate the rhyme distributions for the poetry specific models, unconditioned and conditioned LLMs, respectively.

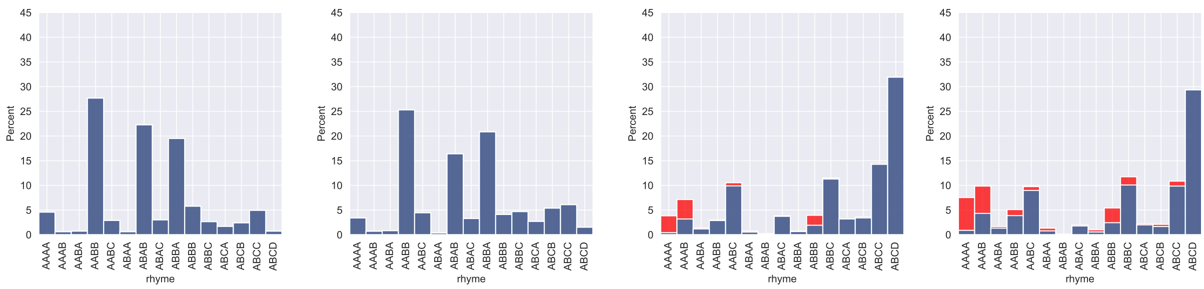
Best model We rank the models for each dimension and then average the ranks across the five dimensions to determine the overall rankings. For dimensions with multiple metrics, such as the three memorization metrics (due to different evaluation levels) and the three lexical metrics (measuring local or global lexical diversity), we first rank the models according to each metric and then average these ranks to represent that dimension. The results are shown in Table 8 and 9 for German and English respectively.

L	model	h	m	M	μ	σ	std
de	HUMAN	1.00	4	65	24.40	23	6.39
de	*DeepSpeare*	0.63	14	30	21.69	22	2.45
de	*SA*	0.88	10	44	24.44	24	5.36
de	$ByGPT5_S$	0.84	9	43	22.11	22	4.86
de	$ByGPT5_L$	0.79	9	40	21.09	21	4.59
de	$GPT2_S$	0.59	9	32	19.18	19	3.54
de	$GPT2_L$	0.73	13	41	21.98	22	3.55
de	$LLaMA2_S$	0.57	9	31	18.84	19	3.29
de	$LLaMA2_L$	0.55	9	30	18.73	19	3.17
de	*LLaMA3*	0.74	12	40	21.39	21	3.99
de	$ByGPT5_S^{con}$	0.82	11	47	22.38	22	4.98
de	$ByGPT5_L^{con}$	0.81	9	45	21.78	21	5.17
de	$GPT2_S^{con}$	0.70	11	37	20.68	20	3.56
de	$GPT2_L^{con}$	0.79	14	45	24.14	24	4.38
de	$LLaMA2_S^{con}$	0.83	12	49	24.22	23	5.41
de	$LLaMA2_L^{con}$	0.62	12	34	20.18	20	2.84
de	$LLaMA3^{con}$	0.76	10	47	21.69	21	4.14
en	HUMAN	1.00	4	67	28.06	28	6.26
en	*DeepSpeare*	0.57	15	33	23.85	24	2.85
en	*SA*	0.92	12	52	27.36	27	5.38
en	$ByGPT5_S$	0.80	12	44	25.30	25	5.09
en	$ByGPT5_L$	0.77	11	47	24.97	25	4.87
en	$GPT2_S$	0.69	13	55	24.11	24	4.48
en	$GPT2_L$	0.72	13	56	24.74	24	4.94
en	$GPTNeo_S$	0.55	11	55	22.67	22	3.89
en	$GPTNeo_L$	0.48	13	34	21.93	22	3.16
en	$LLaMA2_S$	0.87	15	75	28.60	27	7.52
en	$LLaMA2_L$	0.67	12	54	23.95	24	4.50
en	*LLaMA3*	0.59	14	60	23.20	23	4.23
en	$ByGPT5_S^{con}$	0.85	13	42	26.21	26	4.96
en	$ByGPT5_L^{con}$	0.84	14	42	25.85	25	4.84
en	$GPT2_S^{con}$	0.86	17	61	28.37	27	6.18
en	$GPT2_L^{con}$	0.83	16	70	27.82	27	6.15
en	$GPTNeo_S^{con}$	0.74	16	49	25.13	24	4.47
en	$GPTNeo_L^{con}$	0.53	12	35	22.26	22	3.36
en	$LLaMA2_S^{con}$	0.70	17	74	33.55	32	7.83
en	$LLaMA2_L^{con}$	0.81	15	56	26.92	26	5.80
en	$LLaMA3^{con}$	0.78	16	65	27.12	26	5.35

Table 6: Reported statistical and distance measures regarding the length of training data and generated quatrains. h = histogram intersection score between sample and training data, μ = mean length, σ = median, std = standard deviation, m = minimal length, M = maximal length.

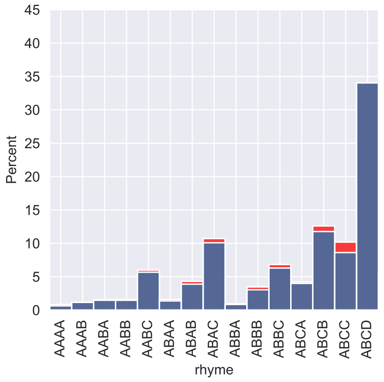
	DE		EN	
Model	Entropy	KL Divergence	Entropy	KL Divergence
HUMAN	2.90	0.00	3.10	0.00
DeepSpeare	2.97	**0.55**	3.16	0.48
SA	3.14	$\underline{1.43}$	3.22	1.17
$\mathit{ByGPT5}_L$	2.89	1.23	2.92	1.08
$\mathit{ByGPT5}_S$	3.13	1.09	2.91	1.13
$\mathit{GPT2}_L$	2.86	1.26	2.97	1.06
$\mathit{GPT2}_S$	3.16	1.13	2.99	1.03
GPTNeo_L	-	-	2.80	$\underline{1.18}$
GPTNeo_S	-	-	3.16	0.96
$\mathit{LLaMA2}_L$	2.93	1.18	3.24	0.71
$\mathit{LLaMA2}_S$	3.18	1.04	3.24	0.71
LLaMA3	3.27	0.83	3.45	0.56
$\mathit{ByGPT5}_L^{con}$	3.17	0.67	3.22	0.83
$\mathit{ByGPT5}_S^{con}$	3.16	0.58	3.38	0.54
$\mathit{GPT2}_L^{con}$	2.98	0.99	3.41	0.61
$\mathit{GPT2}_S^{con}$	3.11	1.04	3.22	0.85
GPTNeo_L^{con}	-	-	3.43	**0.45**
GPTNeo_S^{con}	-	-	3.29	0.83
$\mathit{LLaMA2}_L^{con}$	2.69	1.33	2.89	0.95
$\mathit{LLaMA2}_S^{con}$	3.11	0.71	2.67	1.07
$\mathit{LLaMA3}^{con}$	2.98	1.06	2.58	0.94

Table 7: Entropy and KL divergence of rhyme distributions. We bold the lowest and underline the highest KL divergence from human to model distributions.

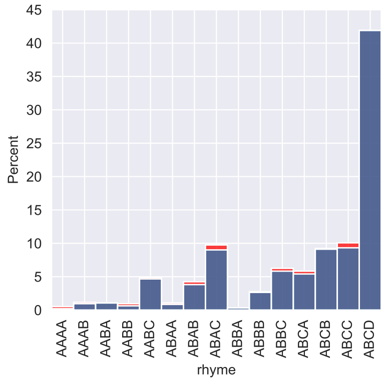


(a) *DeepSpeare* (de) (b) *DeepSpeare* (en) (c) *SA* (de) (d) *SA* (en)

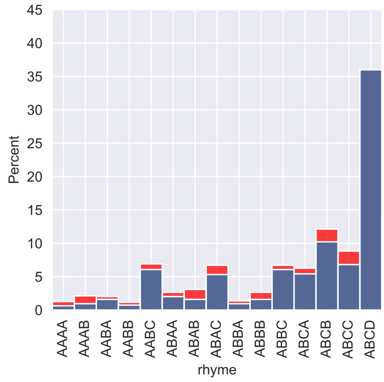
Figure 4: Distribution of rhyme schemes in the samples from *DeepSpeare* and *SA* models for German and English.



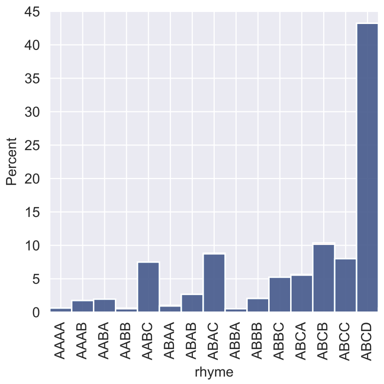
(a) *ByGPT5*$_S$



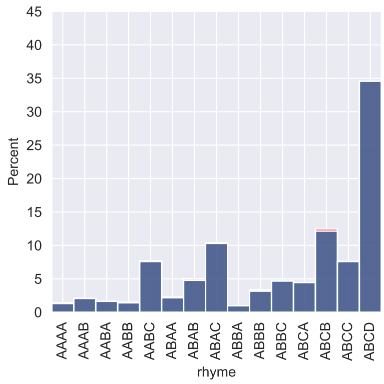
(b) *ByGPT5*$_L$



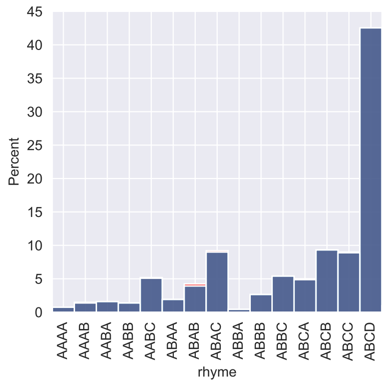
(c) *GPT2*$_S$



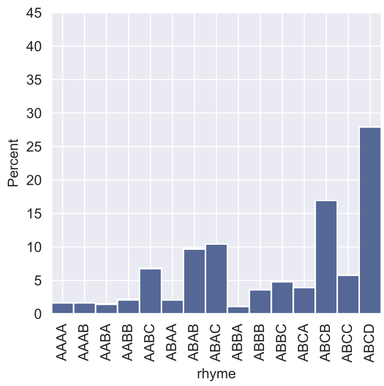
(d) *GPT2*$_L$



(e) *LLaMA2*$_S$

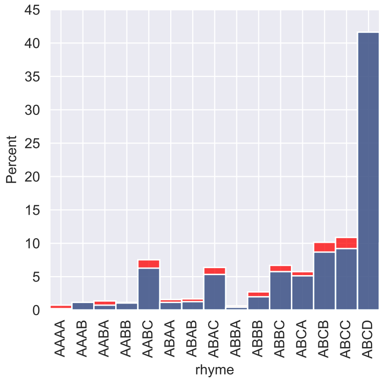


(f) *LLaMA2*$_L$

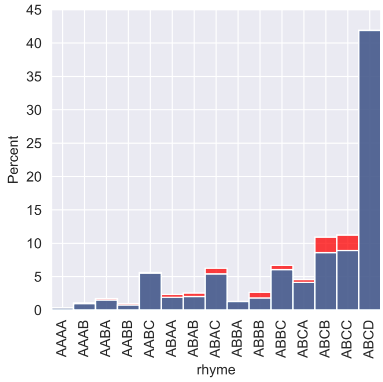


(g) *LLaMA3*

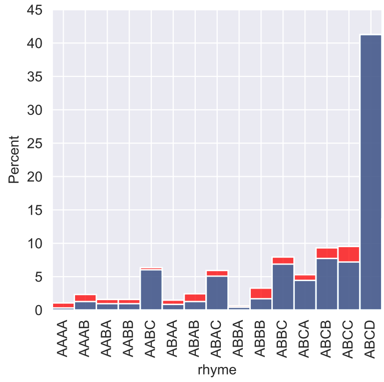
Figure 5: Rhyme distribution plots for samples generated by **German unconditioned** large language models.



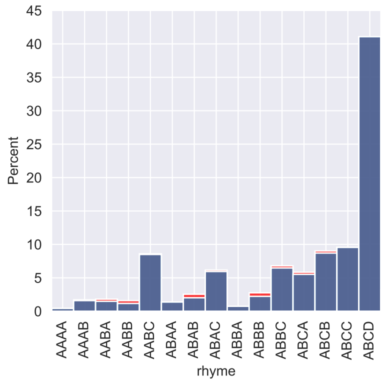
(a) *ByGPT5$_S$*



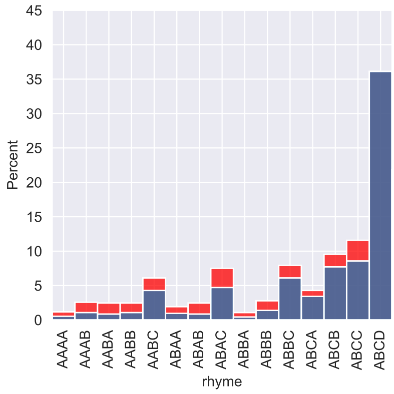
(b) *ByGPT5$_L$*



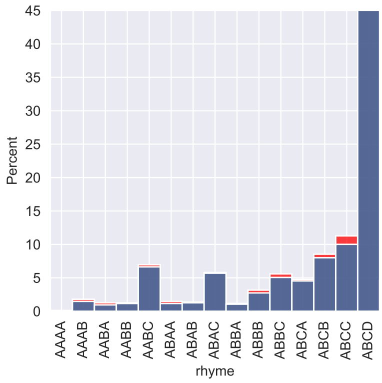
(c) *GPT2$_S$*



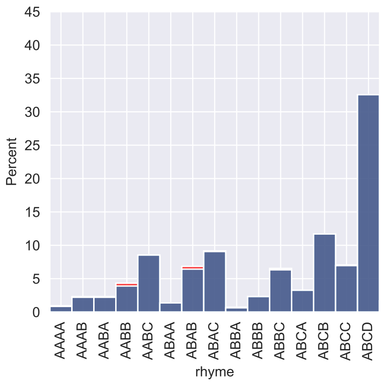
(d) *GPT2$_L$*



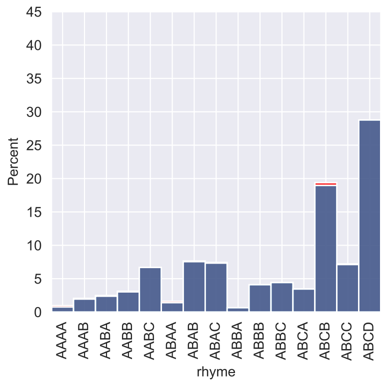
(e) *GPTNeo$_S$*



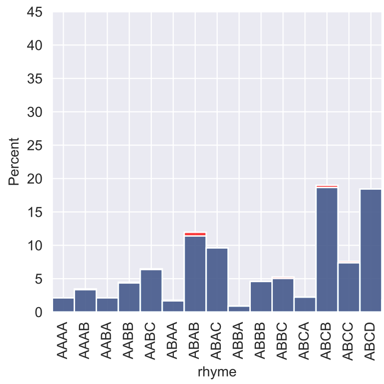
(f) *GPTNeo$_L$*



(g) *LLaMA2$_S$*

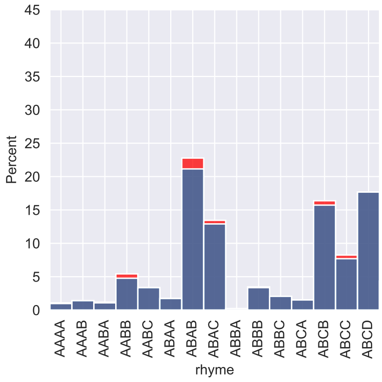


(h) *LLaMA2$_L$*

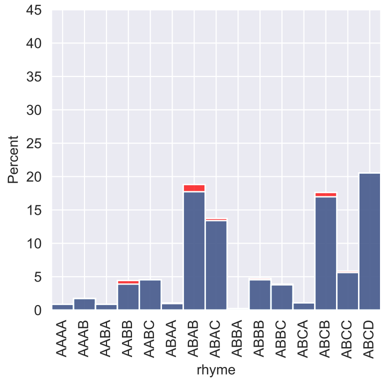


(i) *LLaMA3*

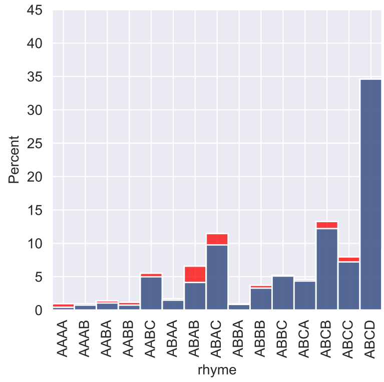
Figure 6: Rhyme distribution plots for samples generated by **English unconditioned** large language models.



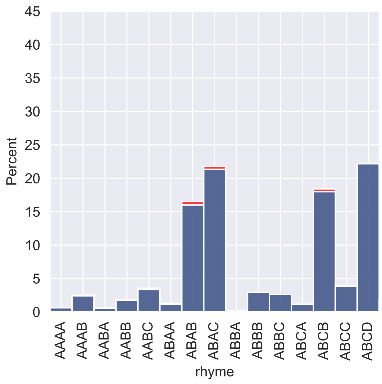
(a) *ByGPT5*$_S^{con}$



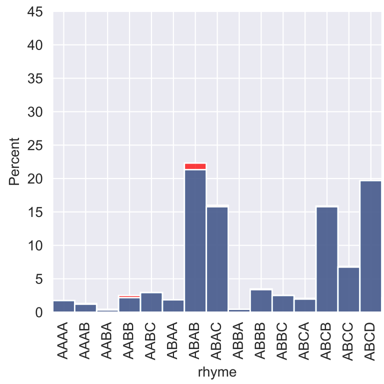
(b) *ByGPT5*$_L^{con}$



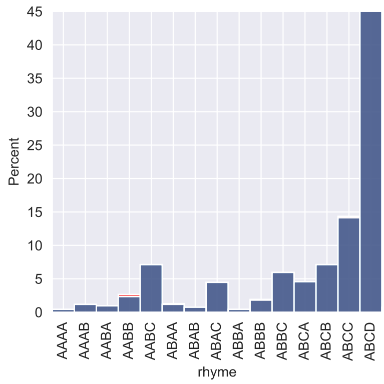
(c) *GPT2*$_S^{con}$



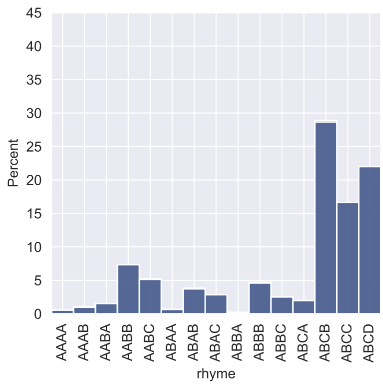
(d) *GPT2*$_L^{con}$



(e) *LLaMA2*$_S^{con}$

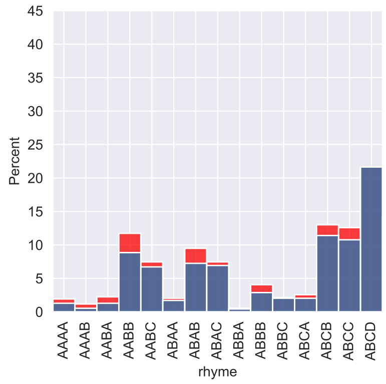


(f) *LLaMA2*$_L^{con}$

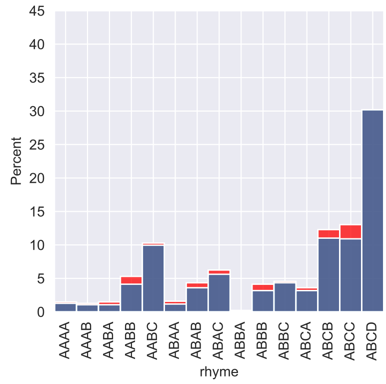


(g) *LLaMA3*^{con}

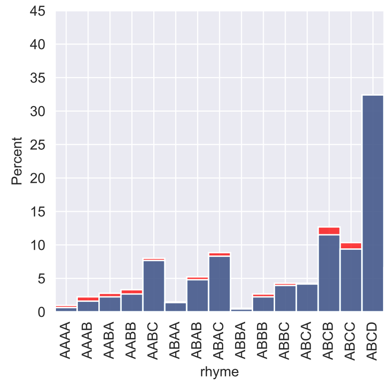
Figure 7: Rhyme distribution plots for samples generated by **German conditioned** large language models.



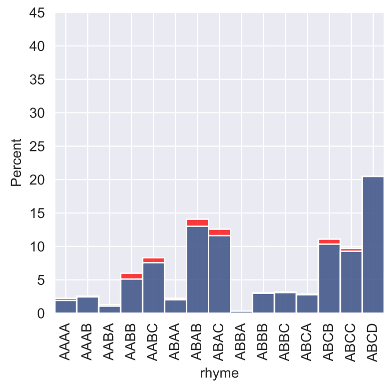
(a) *ByGPT5*$_S^{con}$



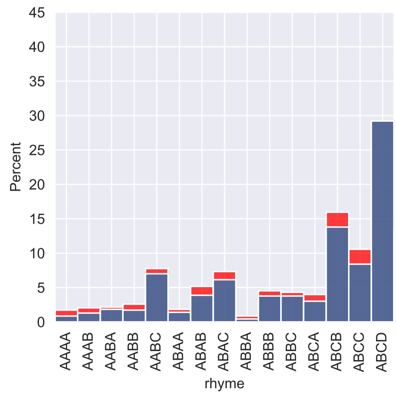
(b) *ByGPT5*$_L^{con}$



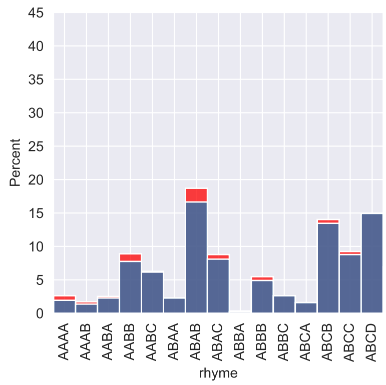
(c) *GPT2*$_S^{con}$



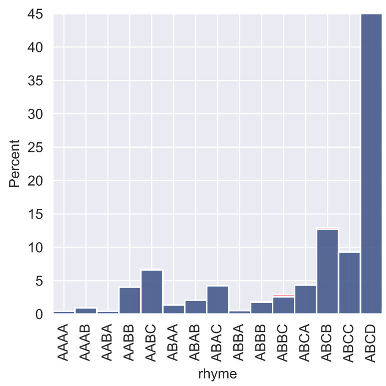
(d) *GPT2*$_L^{con}$



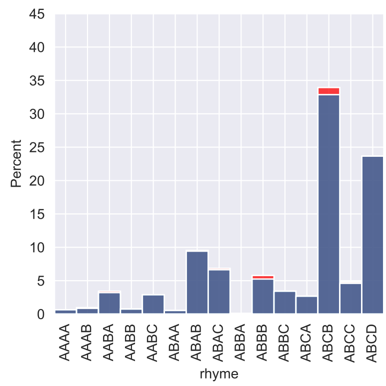
(e) *GPTNeo*$_S^{con}$



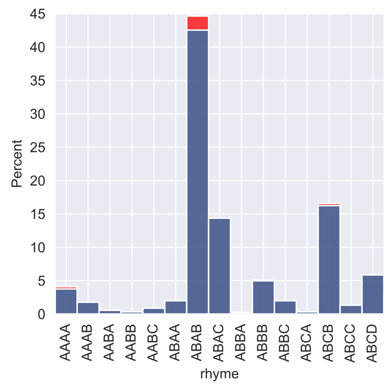
(f) *GPTNeo*$_L^{con}$



(g) *LLaMA2*$_S^{con}$



(h) *LLaMA2*$_L^{con}$



(i) *LLaMA3*^{con}

Figure 8: Rhyme distribution plots for samples generated by **English conditioned** large language models.

Language	Model	Size	Conditioned	semantic	lexical	length	rhyme	memorization	avg_rank
de	BYGPT5	L	TRUE	2.0	4.0	5.0	3.0	1.7	3.1
de	BYGPT5	S	TRUE	3.5	6.0	4.0	2.0	1.3	3.4
de	2SA	-	-	1.0	2.7	1.0	16.0	2.0	4.5
de	1DS	-	-	5.0	10.3	12.0	1.0	1.0	5.9
de	BYGPT5	S	FALSE	6.0	11.0	2.0	10.0	2.7	6.3
de	BYGPT5	L	FALSE	4.0	8.3	6.0	13.0	3.0	6.9
de	LLAMA3	-	FALSE	9.5	6.3	9.0	5.0	6.0	7.2
de	LLAMA3	-	TRUE	6.5	7.3	8.0	9.0	5.7	7.3
de	LLAMA2	S	TRUE	13.5	13.0	3.0	4.0	4.0	7.5
de	GPT2	L	TRUE	12.5	4.7	7.0	6.0	8.3	7.7
de	LLAMA2	L	FALSE	9.5	2.7	16.0	12.0	5.3	9.1
de	LLAMA2	S	FALSE	8.0	10.0	15.0	8.0	5.0	9.2
de	GPT2	L	FALSE	14.0	5.7	10.0	14.0	8.7	10.5
de	GPT2	S	TRUE	15.0	15.0	11.0	7.0	6.3	10.9
de	LLAMA2	L	TRUE	12.5	13.0	13.0	15.0	8.0	12.3
de	GPT2	S	FALSE	13.5	16.0	14.0	11.0	7.7	12.4

Table 8: Ranking of **German** models for each dimension, as well as the average ranks across all dimensions.

Language	Model	Size	Conditioned	semantic	lexical	length	rhyme	memorization	avg_rank
en	BYGPT5	S	TRUE	3.5	11.7	4.0	3.0	2.0	4.8
en	2SA	-	-	1.0	4.0	1.0	19.0	1.0	5.2
en	BYGPT5	L	TRUE	2.0	9.7	5.0	9.0	1.7	5.5
en	1DS	-	-	3.5	9.0	17.0	2.0	2.3	6.8
en	LLAMA2	S	FALSE	17.5	5.7	2.0	6.0	4.7	7.2
en	LLAMA3	-	TRUE	12.0	1.7	9.0	11.0	3.3	7.4
en	GPT2	L	TRUE	9.0	9.0	6.0	5.0	9.3	7.7
en	LLAMA2	L	TRUE	12.0	5.0	7.0	12.0	4.0	8.0
en	LLAMA2	S	TRUE	7.0	3.3	13.0	16.0	1.3	8.1
en	LLAMA3	-	FALSE	13.0	3.0	16.0	4.0	9.0	9.0
en	LLAMA2	L	FALSE	9.0	6.3	15.0	7.0	10.3	9.5
en	GPT2	S	TRUE	17.5	14.0	3.0	10.0	3.7	9.6
en	BYGPT5	L	FALSE	5.5	15.7	10.0	17.0	3.0	10.2
en	BYGPT5	S	FALSE	5.5	17.3	8.0	18.0	2.7	10.3
en	GPTNEO	L	TRUE	13.5	13.0	19.0	1.0	10.0	11.3
en	GPTNEO	S	TRUE	16.0	17.0	11.0	8.0	5.7	11.5
en	GPT2	L	FALSE	10.5	11.0	12.0	15.0	11.3	12.0
en	GPT2	S	FALSE	17.0	19.0	14.0	14.0	11.7	15.1
en	GPTNEO	S	FALSE	17.5	20.0	18.0	13.0	12.0	16.1
en	GPTNEO	L	FALSE	17.5	14.7	20.0	20.0	11.3	16.7

Table 9: Ranking of **English** models for each dimension, as well as the average ranks across all dimensions.