Evaluating Diversity in Automatic Poetry Generation

Anonymous EACL 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) and types of fine-tuning (conditioned vs. unconditioned). We find that current automatic poetry systems are considerably underdiverse along all dimensions — they tend to memorize, 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. Concerning general purpose LLMs, some of them exhibit very high risk of memorization — an extreme form of lack of diversity — and this depends on the size of the training data set, the size and type of the LLM, and the type of training, as we show. 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).

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 dialog generation and image captioning, assessing lexical overlaps in n-best NLG outputs for the same input. *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, where the goal is to avoid "safe and bland" responses that "average out" the sentences observed in the training set. Here, semantic diversity has been measured, e.g., with the help of embedding-based similarity (Du and Black, 2019).

In our work on diversity in poetry generation, we complement these 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. As our dataset, we use QuaTrain (Belouadi and Eger, 2023) consisting of quatrains (in English and German). We describe it in more detail in §5.1 below.

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. Therefore, we consider a low or minimal degree of memorization as a pre-requisite for diversity and analyze the portion of generated poems that are (near-)copies from the training data. 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. 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. 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.

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 Models

Our experiments use 2 different model classes.

4.1 Poetry-specific models

Deepspeare. 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.

4.2 General purpose LLMs

All models in this category are decoder-only transformer architectures. In our experiments, we train them in an unconditioned and style-conditioned manner (see Section 5.2).

GPT2 GPT2 (Radford et al., 2019) is the last GPT model made publicly available. It is a large word level transformer-based language model pretrained on approximately 40 GB of text. Four different model versions were released, with the number of parameters ranging from 125 million to 1.5 billion for the largest. In this work, we deploy two model versions: GPT2-small (125M parameters) and GPT2-large (774M parameters) for both English and German.

GPTneo GPTneo (Black et al., 2022) is an open-source token level LLM by EleutherAI (`https://www.eleuther.ai/`) with the aim to provide publicly available replications of GPT3. It is pretrained on 825 GB of text data. Currently, four versions have been released, with the number of parameters ranging from 125 million up to 20 billion. We deploy GPTneo-small and GPTneo-xl with 125M and 1.3B parameters for English. GPTneo is not available for German.

ByGPT5 ByGPT5 (Belouadi and Eger, 2023) is a decoder-only character level LLM based on the encoder-decoder character level model byT5 (Xue et al., 2022) where the encoder part of byT5 is completely removed, reducing the number of parameters by 75%. The remaining decoder-only model is then pretrained using OpenWebText for English (38GB text data) and CC100 (Conneau et al., 2020) (67GB text data) for German. Three versions are released for both English and German, with model sizes ranging from 73 to 298M parameters. We use ByGPT5-base (140M params) and ByGPT5-medium (290M) for both English and German.

5 Experimental Setup

5.1 Training Data

We use QuaTrain, a large dataset of quatrains published by Belouadi and Eger (2023). It consists of English and German quatrains and has been generated by aggregating different publicly available poetry datasets. QuaTrain 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. QuaTrain is automatically annotated for meter and rhyme using high-quality classifers (especially for rhyme). Table 1 provides basic information about the size of the dataset.

	English	German
# Quatrains	662,877	1,483,785

Table 1: Size of training data sets.

5.2 Training

Deepspeare. Deepspeare leverages pretrained static word vectors. We use QuaTrain to train our own English and German word embeddings using Word2vec (Mikolov et al., 2013), training word embeddings with a dimension of 100 and a window size of 5. As Deepspeare is designed to process sonnet data during training, we use training data to create artificial sonnets. Thus, we concatenate three quatrains and append one couplet that we get from an additional dataset (partially contained in QuaTrain) called PoeTrain[1]. We split the training data into a train, test, and validation set using a ratio of 80 to 10 to 10 (the latter two are used to measure losses each epoch), training for 10 epochs.

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]; 2) training the LM component; 3) training a final aggregated model in a generative adversarial setup. We train this final model for 10 epochs. As we encounter different errors when trying to train a German version, we use the English variant only.

Unconditioned LLMs. In this setup, we fine-tune our decoder-only LLMs in an *unconditioned* manner: we process quatrains during training without passing any information about rhyme (or meter).

[1] `https://github.com/potamides/uniformers/blob/main/uniformers/datasets/poetrain/poetrain.py` Analyses show that QuaTrain contains 0.4% of English and 66% of German PoeTrain data. Therefore, English sonnets receive ∼14% and German sonnets ∼5% additional data.

[2] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

We split training data into a train and validation set using a ratio of 90 to 10. All models except GPTneo (being available only in English) are trained both in English and German. We fine-tune all models (English and German) for 10 epochs.

Style-conditioned LLMs. In contrast to unconditioned training, we provide information about rhyme (and meter) by prepending special style tokens to each quatrain during training. This follows the setup of Belouadi and Eger (2023) and makes models *explicitly* aware of different rhyme schemes. As for the unconditioned variants, all models except GPTneo are trained in English and German. We use the same validation split and again fine-tune each model for 10 epochs.

Summary. We end up with 23 models that can be assigned to three categories: 1) **Poetry specific** LSTM-based models (Deepspeare and SA). Besides a language model part, these models incorporate additional specialized components to handle poetry-specific stylistic features such as rhyme. We have three models in total for English and German. 2) **Unconditioned LLMs** (transformer-based decoder-only general purpose LLMs). These models do not possess any specialized architecture for poetry. No information about meter or rhyme has actively been passed during training. We have two subcategories: word and character level models. The first group (GPT2, GPTneo) processes data on the word/subword level. ByGPT5 represents the character-level group. We have 10 models in total (6 English and 4 German ones). 3) **Style-conditioned LLMs**. These have the same architecture, models, and subgroups as category 2. Information about rhyme (and meter) is passed in the form of special tokens during training (only). In order to distinguish between unconditioned and style-conditioned model variants, we append the prefix "poetry" to style-conditioned models.

Table 5 (appendix) provides an overview of all models belonging to the second and third category (transformer-based LLMs).

5.3 Sampling

From each model class, we randomly draw 500 generated poems. Whenever wo do a direct comparison between training and generated data (e.g. when comparing lexical diversity), we randomly draw 10 samples of size 500 (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. We mainly provide results for samples obtained via standard sampling. However, we briefly discuss the effects of sampling and search during decoding in Section 7.

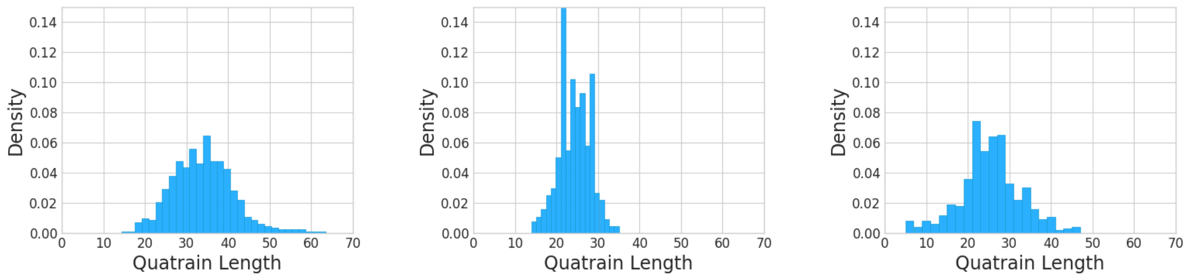
6 Experiments and Results

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.

Model	EN	DE
poetry-GPT2-small	0.010	0.002
poetry-GPT2-large	0.806	0.094
poetry-GPTneo-small	0.141	-
poetry-GPTneo-xl	0.886	-
poetry-byGPT5-base	0.000	0.002
poetry-byGPT5-medium	0.006	0.048
poetry-GPT2-large (660k)	0.806	0.822

Table 2: Memorization rates in samples generated by the listed models.

Memorization Table 2 shows the calculated memorization scores for samples from a subset of our models. Our **poetry-specific** LSTM models show no memorization. **Unconditioned LLMs** exhibit similar results. The only model slightly affected is the large English version of GPT2, with a score of 0.2%. Thus, we omit all these results from the table. However, the third category of **style-conditioned LLMs** reveals remarkable differences, with memorization scores ranging from 0% to 88%. Within each model family, the memorization rate for larger models is strictly higher compared to smaller ones. The strength of this correlation not only varies across model families, but also appears to depend on the language: the memorization rates for the English GPT2 variants show a substantial increase from 1% (small) to approximately 80% (large), while the rates for the German models experience a smaller increase, from 0.2% to below 10%. Models of the GPTneo family generally show the highest memorization values, with 14% for the small variant (the highest value of all small models) and 88% for the XL variant (with 1.3B parameters the by far largest model in our collection). The memorization rates of the character-level ByGPT5 models are remarkably low comparatively. The English base variant of ByGPT5 is the



(a) QuaTrain (EN) (b) Deepspeare (EN) (c) ByGPT5-medium (EN)

Figure 1: Length distribution of human poems (left), Deepspeare (middle) and ByGPT5-base (right) for English.

only style-conditioned model that has a score of 0. The medium English model shows a score of 0.6%. Memorization rates for the German models increase from 0.2% to roughly 5%, representing the second-smallest rise observed.

Analysis: Since our German and English data vary vastly in size, we reduce the size of the German training data set to fit the size of the English training data (we randomly select 660k German quatrains) to see its effect on memorization and retrain poetry-GPT2-large on it, which had around 80% memorization for English but less than 10% for German. On the reduced size of the training data set, the German model has now similar memorization as the English model (see results below dashed line in Table 2). This indicates that the memorization rates are not language dependent but depend on model and training data size: larger models trained on less data memorize more. Examples for different levels of memorization are provided in Tables 9, 10 and 11 in the appendix.

Length Table 6 (appendix) reports statistics on the length of poems, both human and automatically generated.

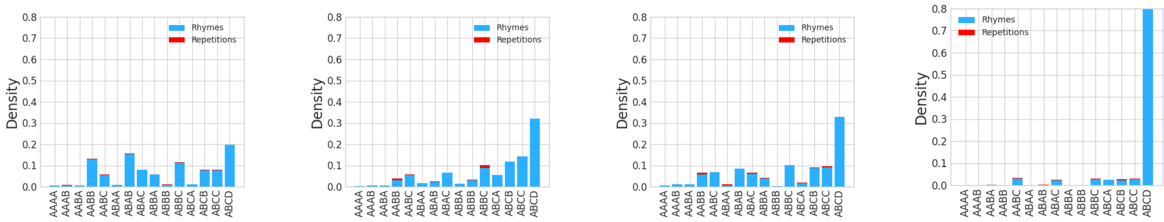
Humans poems in English have on average 34 tokens, while German poems have 25 tokens. The histogram intersection values of different models with human poems range from 0.04 (poetry-GPT2-small German) to 0.95 (GPT2-small German) — it is remarkable that style conditioning worsens the match so much for this model. The character-level LLMs — variants of ByGPT5 — fit the human distribution the best on average, independent of whether the model is trained with style-conditioning or not. The poetry-specific Deepspeare model matches the human distribution worst: the generated poems are too short and too underdiverse (in terms of standard deviation). Models typically fit the German distribution, with more training data, better. Figure 1 illustrates the length distribution of human poems, Deepspeare and ByGPT5-medium for English.

Rhyme Figure 2 (a) shows the distributions of rhyme schemes in our human training datasets (exemplarily for German), while Table 8 shows the corresponding numerical values. Most rhymes in the training data are classified as real rhymes. For both languages, roughly 20% of all quatrains in training do not rhyme at all (rhyme scheme ABCD). Excluding ABCD, the top 3 dominant rhyme schemes by appearance are AABB, ABAB and ABBC for both datasets, with a total share of approximately 40% in each language, and all between 10-20%.

Poetry-specific models: Figure 4 (appendix) shows the distributional plots for Deepspeare and SA. We see that ABCD dominates throughout all samples, with portions of roughly 45% for the English models and approximately 25% for the German version of Deepspeare, which means that these models achieve a lower diversity in their rhyme patterns compared to human data. Besides ABCD, no other rhyme patterns dominate, the most frequent non-ABCD rhyme schemes typically make up less than 10% of all schemes.

Figures 5 and 6 (appendix) show the distributions of rhyme patterns for **unconditioned LLMs**. For unconditional LLMs, the distributions are even more skewed towards the ABCD scheme (clearly above 50% and even above 70% for word-level models), suggesting that these models are even more incapable of learning the concept of rhyming. While models of the ByGPT5 family rhyme better, they also have more repetitions, with the English base version and the medium German version being affected the most.

Style-conditioned LLMs are shown in Figures 7 and 8 (appendix). They achieve better diversity



(a) QuaTrain (DE) (b) poet.-GPT2-small (DE) (c) poet.-ByGPT5-med. (DE) (d) GPT2-large (DE)

Figure 2: Distribution of rhyme schemes in the training data (a) and generated samples (b,c,d).

	ATTR ↑	MATTR ↑	MTLD ↑
QuaTrain	0.865	0.861	138.57
Deepspeare	**0.915**	**0.883**	**151.21**
SA	0.880	0.861	118.14
GPT2-small	0.872	0.728	45.33
GPT2-large	0.858	0.737	47.66
GPTneo-small	0.862	0.745	49.05
GPTneo-xl	0.844	0.705	40.45
byGPT5-base	0.804	0.794	64.68
byGPT5-med.	0.879	0.812	86.60
poet.-GPT2-small	0.856	0.848	112.65
poet.-GPT2-large	0.860	0.857	124.82
poet.-GPTneo-small	0.864	0.859	131.03
poet.-GPTneo-xl	0.856	0.854	130.30
poet.-byGPT5-base	0.846	0.842	112.00
poet.-byGPT5-med.	0.850	0.843	108.99

Table 3: Lexical diversity measures for English models.

in their generated rhyme patterns: only between 25-40% of all rhyme patterns are ABCD. The larger word-level models even achieve roughly 15% for the pattern AABB in English, which is close to the value of the human poems — but recall that these models also memorize a lot more. Like the human data, the large word-level and the ByGPT5 models peak for the patterns AABB and ABBC in English. The ByGPT5 models in German also have similar peaks as the human training data, namely, AABB, ABAB and ABBC.

Comparing the three model categories, poetry-specific models and style-conditioned LLMs show some rhyming ability. The unconditioned LLMs reveal clear deficits, as most quatrains do not rhyme at all. GPT2 and GPTneo perform worse than ByGPT5. All models have clearly higher ABCD rhyming schemes than the human data, thus are underdiverse concerning rhyming.

Figure 2 gives an illustrative comparison of the rhyme distributions in the human data vs. poetry-GPT2-small, poetry-ByGPT5-medium and GPT2-large for German, showcasing different forms of distributions.

Lexical Diversity. Table 3 shows the lexical diversity results for English data, and Table 7 (appendix) for German. For English, the least lexically diverse poems according to the 2 global diversity metrics (MATTR, MTLD) are **unconditioned LLMs**. The MTLD metric of global diversity is particularly low for these models. The local diversity of unconditioned LLMs, on the other hand, is close to human performance, or even exceeds it for GPT2-small and ByGPT5-med. 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. **Style-conditioned LLMs** and particularly the **poetry-specific** Deepspeare are much more diverse at the local and global level; the latter even beats human poems in terms of diversity. This is slightly surprising, but may be explained by the fact that human poetry often contains forms of parallelism or redundancy, such as repetitions, which "normal" text does not contain. The fact that generated poems of some model classes are more lexically diverse than human poems is also an indication that measuring lexical diversity is not sufficient for poetry.

Semantic Similarity Table 4 presents statistical indicators for the semantic similarity of quatrains: (i) between generated samples and the human data, (ii) within model generated samples, (iii) within the human training data.

For both English and German, the human data is most diverse with respect to semantic similarity; the average maximum 'within' values are 0.46 or lower. Otherwise, we observe similar trends as for lexical diversity: **Unconditioned LLMs** are least diverse wrt. semantic similarity (0.46-0.73), but the character-level models perform clearly better (0.46-0.55); **style-conditioned LLMs** (0.42-0.55) and **poetry-specific models** (0.44-0.53) produce more

Model	across (↓)	within (↓)
QuaTrain		0.42/0.46
Deepspeare	0.56/0.61	0.44/0.47
SA	0.62/-	0.44/-
GPT2-small	0.57/0.63	0.73/0.69
GPT2-large	0.54/0.62	0.71/0.71
GPTneo-small	0.57/-	0.71/-
GPTneo-xl	0.57/-	0.73/-
ByGPT5-base	0.55/0.59	0.46/0.55
ByGPT5-medium	0.56/0.61	0.55/0.55
poet.-GPT2-small	0.59/0.63	0.43/0.55
poet.-GPT2-large	0.90/0.76	0.43/0.50
poet.-GPTneo-small	0.63/-	0.42/-
poet.-GPTneo-xl	0.93/-	0.43/-
poet.-ByGPT5-base	0.59/0.63	0.43/0.47
poet.-ByGPT5-med.	0.59/0.64	0.43/0.46

Table 4: Average maximum semantic similarity values: (i) across models and humans (middle), (ii) within models including the human QuaTrain dataset (right). Each column: EN/DE.

semantically diverse poetry. However, recall that some style-conditioned LLMs produce poetry that is extremely semantically similar to human poems (due to the memorization effect discussed above): particularly larger and non-character-level models fare worse, with 'across' similarity scores with the human data of over 0.9 for English and over 0.75 for German. From the perspective of semantic diversity, poetry-ByGPT5 and DeepSpeare are the best models.

7 Discussion

Sampling/Searching We deploy various decoding strategies to determine to what extent these can alter the various aspects of diversity in the generated poems. We use different combinations of temperature-based sampling (Ackley et al., 1985), Nucleus sampling (Top-p) (Holtzman et al., 2019) and Top-k sampling (Fan et al., 2018) as sampling strategies and further deploy two variants of contrastive search (Su et al., 2022). Results indicate that the various techniques can only slightly increase diversity in one or more aspects. Moreover, aggressive sampling often leads to output degeneration, causing the models to (partially) repeat verses in a quatrain. We provide some examples in the appendix, see Tables 12 to 17 as well as Figures 9 to 14.

Which is the most diverse model? We have seen that unconditioned LLMs exhibit poor results with regard to different dimensions of diversity: they 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 rhyming, semantic variation, and lexical variation but word level style-conditioned models are prone to severe memorization from the training data, in particular when the model is large and the training set is small. Character-level style-conditioned LLMs produce overall best diversity results and do not deteriorate as a function of model/training data size. In terms of diversity, poetry-specific Deepspeare performs similar as character-level LLMs but requires more modeling effort from human experts (e.g., in developing rhyming components).

8 Conclusion

To date, evaluation of automatic poetry generation has almost exclusively focused on human evaluation and forms of the Turing test. Our work 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, confirming previous evaluations of diversity in other NLG tasks (Ippolito et al., 2019; Schüz et al., 2021; Stasaski and Hearst, 2022). 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 distributions, but also higher lexical and semantic diversity. We also find that memorization — a general and widely discussed limitation of LLMs (Carlini et al., 2021) — is a potential issue in poetry generation, especially for certain combinations of model sizes and finetuning schemes, complementing existing studies in this area (Mireshghallah et al., 2022).

We release all code upon acceptance.

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.

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.

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.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169.

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.

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.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

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

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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).

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

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.

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.

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*.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick.

2022. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.

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.

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.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

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*.

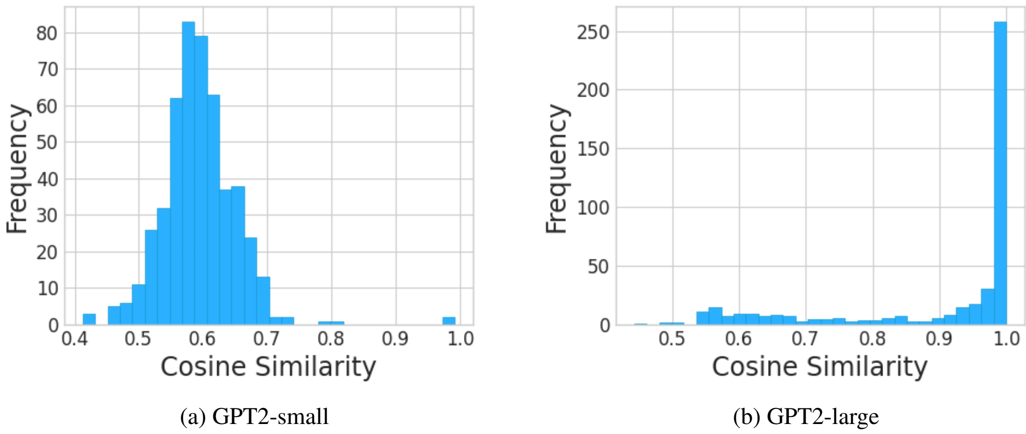
Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, 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 Example Appendix

Model	EN	DE
GPT2-small	124M	124M
GPT2-large	774M	774M
GPTneo-small	125M	-
GPTneo-xl	1.3B	-
byGPT5-base	140M	140M
byGPT5-medium	290M	290M

Table 5: Overview of transformer-based models. Each model is trained both unconditioned and style-conditioned.



(a) GPT2-small

(b) GPT2-large

(c) GPTneo-small

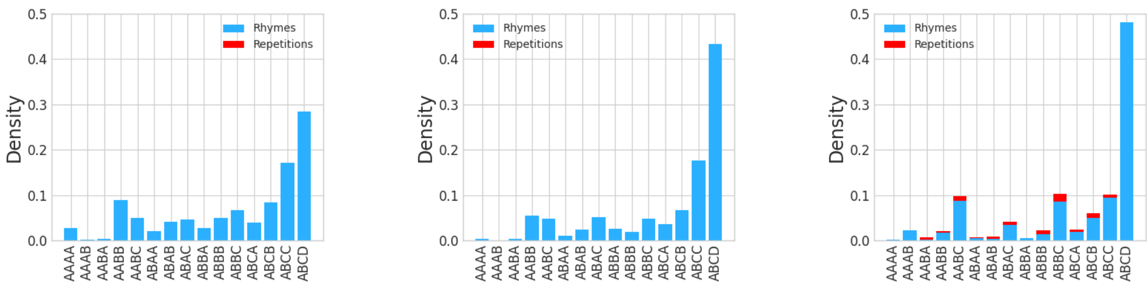
(d) GPTneo-xl

Figure 3: Maximum similarity plots for English style-conditioned GPT2 and GPTneo.

B Decoding methods

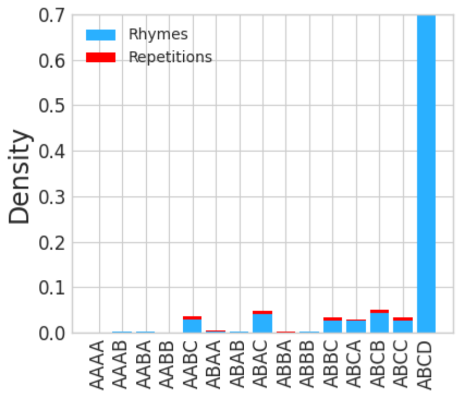
Sample	L.	μ	σ	m	M	h
QuaTrain	EN	34.2	7.6	14	67	1.00
QuaTrain	DE	25.5	7.0	9	48	1.00
Deepspeare	EN	24.2	3.8	14	35	0.37
SA	EN	31.0	5.9	9	47	0.81
Deepspeare	DE	20.0	3.1	12	32	0.56
GPT2-small	EN	28.2	7.1	8	70	0.68
GPT2-large	EN	29.5	8.1	4	68	0.74
GPTneo-small	EN	28.4	7.3	9	59	0.69
GPTneo-xl	EN	26.1	6.6	10	60	0.56
byGPT5-base	EN	29.8	13.6	4	150	0.75
byGPT5-med.	EN	24.8	7.3	5	47	0.51
GPT2-small	DE	24.7	6.8	9	50	0.95
GPT2-large	DE	23.3	6.1	7	45	0.88
byGPT5-base	DE	24.8	6.8	4	52	0.91
byGPT5-med.	DE	26.7	7.4	10	59	0.89
p.-GPT2-small	EN	29.7	5.2	19	52	0.68
p.-GPT2-large	EN	29.7	5.7	16	55	0.67
p.-GPTneo-small	EN	29.0	4.9	17	57	0.63
p.-GPTneo-xl	EN	29.7	5.3	12	51	0.67
p.-byGPT5-base	EN	30.2	6.4	13	63	0.74
p.-byGPT5-med.	EN	29.5	5.9	13	60	0.72
p.-GPT2-small	DE	53.8	6.6	34	78	0.04
p.-GPT2-large	DE	56.4	8.2	34	83	0.06
p.-byGPT5-base	DE	26.7	6.9	11	45	0.93
p.-byGPT5-med.	DE	26.3	6.8	11	45	0.94

Table 6: Reported statistical measures as well as distance measures regarding the length of training data and generated quatrains. std stands for the standard deviation, while m and M denote the minimal and maximal values. h is the histogram intersection score between a sample and the corresponding training data. To facilitate comparison, we draw 10 samples of size 500 from the train set and report mean values

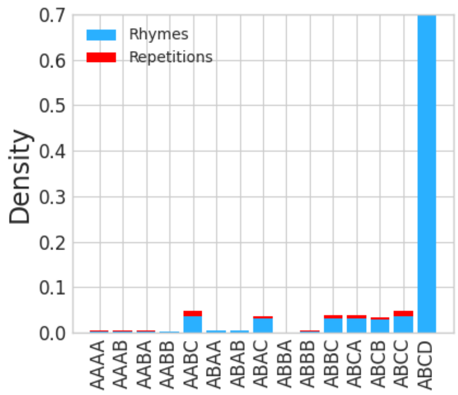


(a) Deepspeare (EN) (b) Deepspeare (DE) (c) Structured Adversary

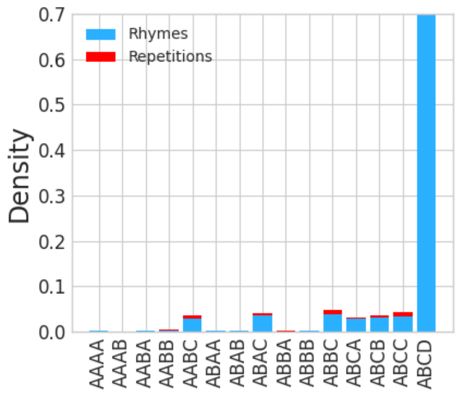
Figure 4: Distribution of rhyme schemes for samples generated by poetry-specific models. Deepspeare vs. Structured Adversary.



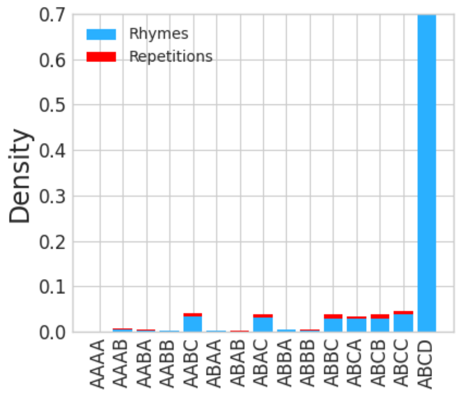
(a) GPT2-small



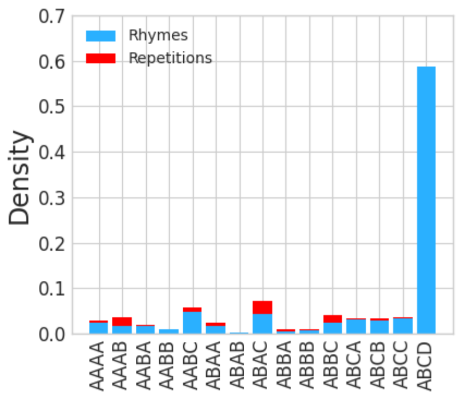
(b) GPT2-large



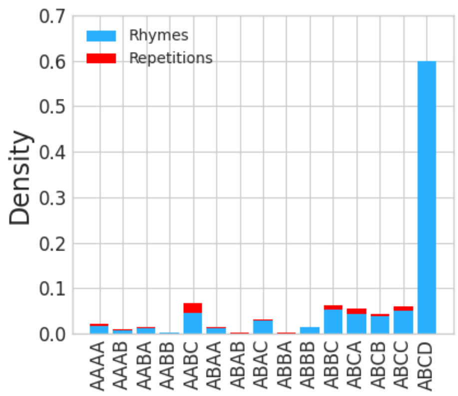
(c) GPTneo-small



(d) GPTneo-xl

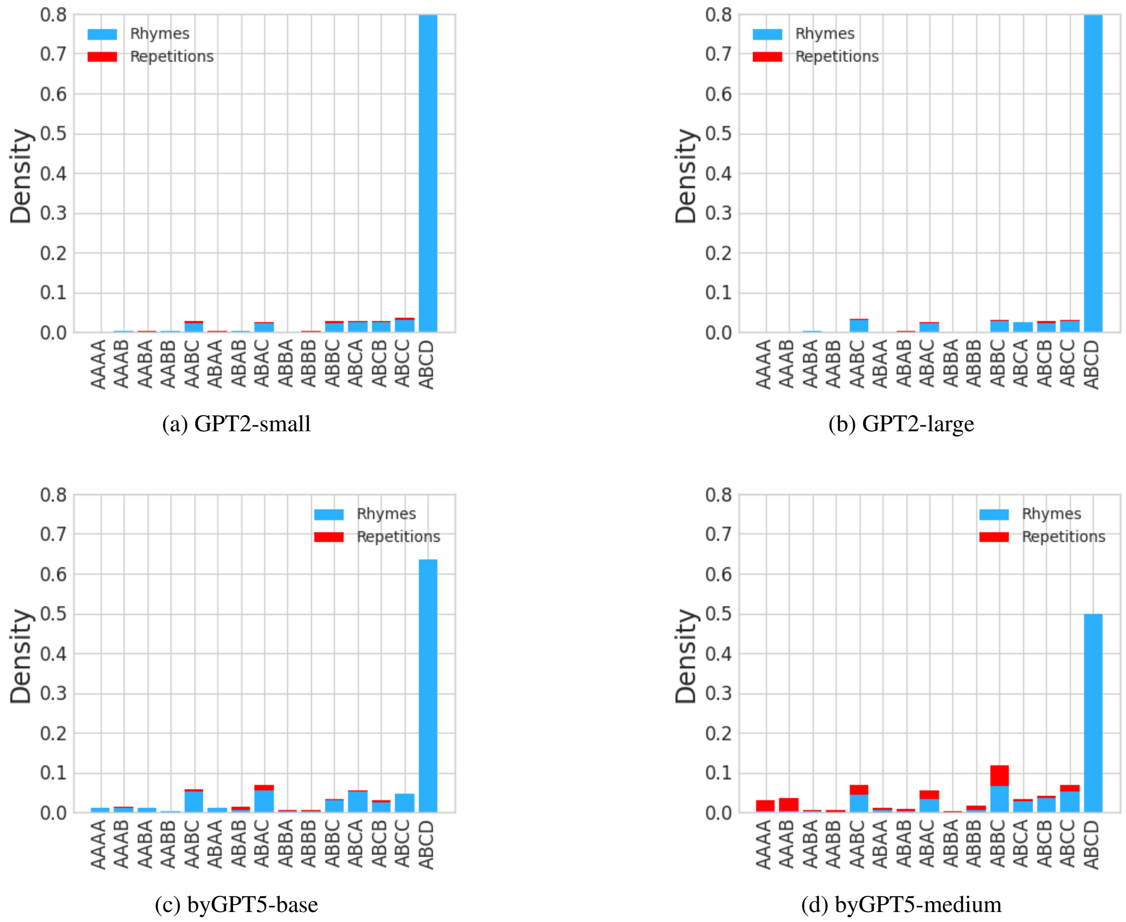


(e) byGPT5-base



(f) byGPT5-medium

Figure 5: Rhyme plots for samples generated by English unconditioned large language models.



(a) GPT2-small

(b) GPT2-large

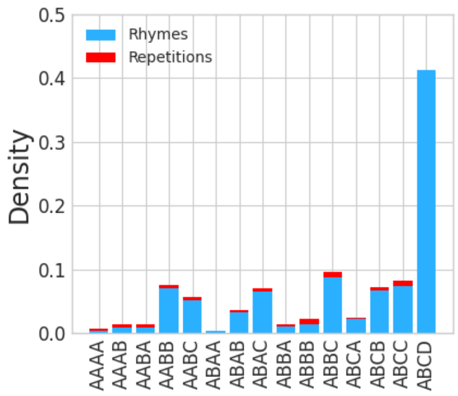
(c) byGPT5-base

(d) byGPT5-medium

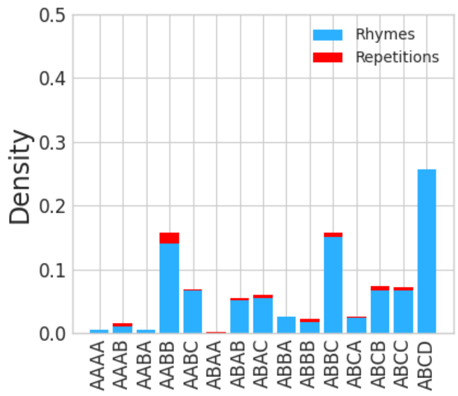
Figure 6: Rhyme plots for samples generated by German unconditioned large language models.

	ATTR ↑	MATTR ↑	MTLD ↑
QuaTrain	0.871	0.854	146.50
DeepSpeare	**0.943**	**0.901**	**203.49**
GPT2-small	0.864	0.721	45.29
GPT2-large	0.882	0.710	41.02
ByGPT5-base	0.863	0.803	84.12
ByGPT5-med.	0.781	0.752	50.93
poetry-GPT2-small	0.805	0.854	164.21
poet.-GPT2-large	0.793	0.839	129.35
poet.-ByGPT5-base	0.860	0.844	120.91
poet.-ByGPT5-med.	0.861	0.844	126.44

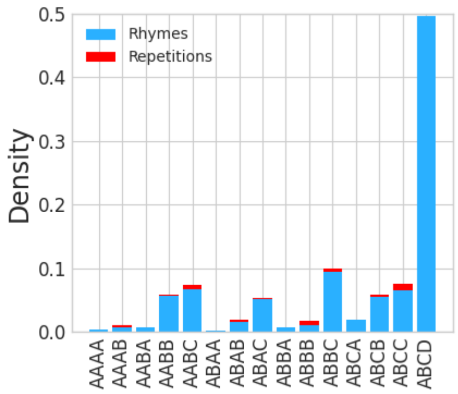
Table 7: German



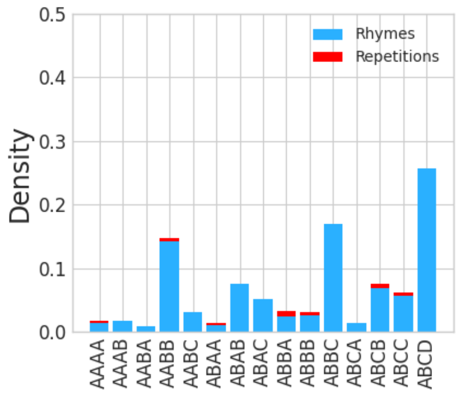
(a) GPT2-small



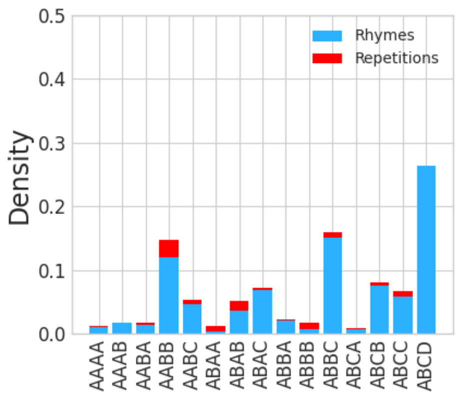
(b) GPT2-large



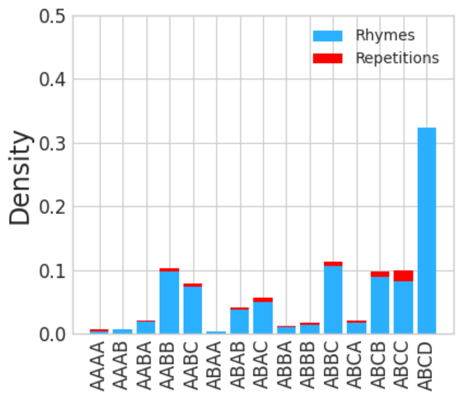
(c) GPTneo-small



(d) GPTneo-xl

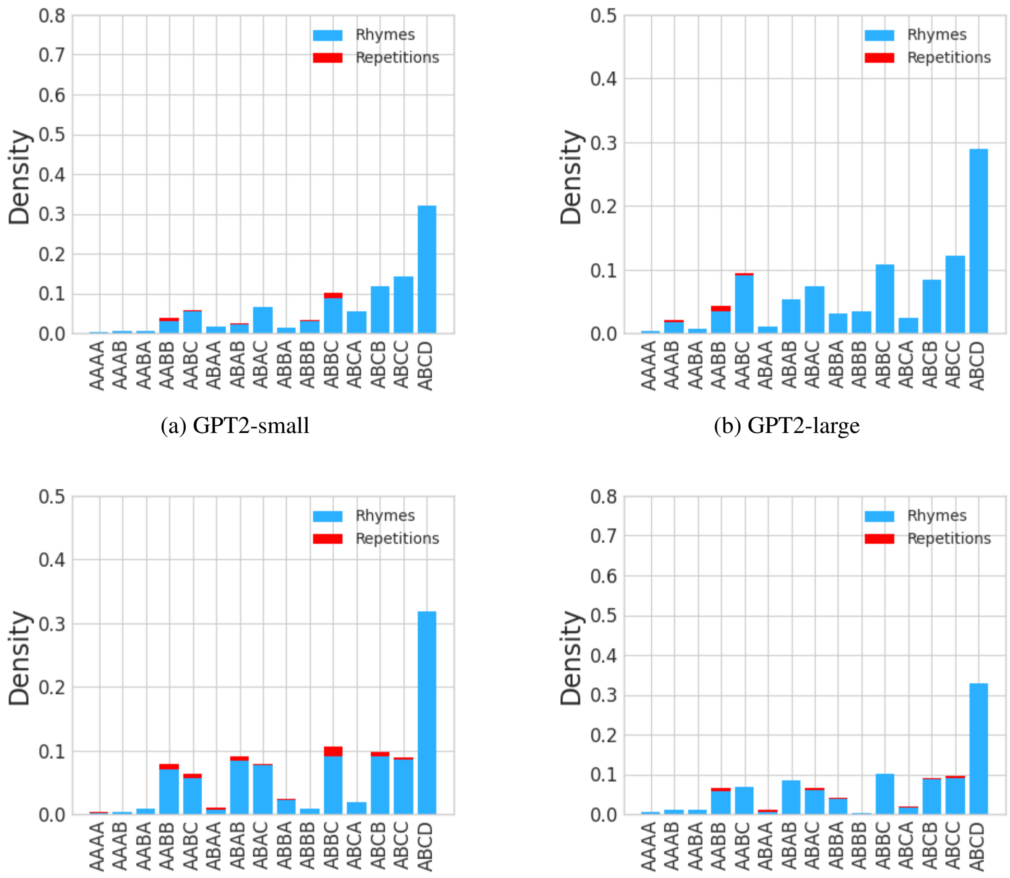


(e) byGPT5-base



(f) byGPT5-medium

Figure 7: Rhyme plots for samples generated by English style-conditioned large language models.



(a) GPT2-small (b) GPT2-large

(c) byGPT5-base (d) byGPT5-medium

Figure 8: Rhyme plots for samples generated by German style-conditioned large language models.

	EN		DE	
Rhyme	Real	Rep.	Real	Rep.
AAAA	0.007	0.002	0.005	0.001
AAAB	0.015	0.002	0.007	0.001
AABA	0.008	0.001	0.006	0.001
AABB	0.183	0.009	0.139	0.002
AABC	0.036	0.002	0.056	0.001
ABAA	0.012	0.002	0.008	0.001
ABAB	0.098	0.008	0.153	0.004
ABAC	0.059	0.004	0.079	0.002
ABBA	0.037	0.002	0.057	0.001
ABBB	0.017	0.002	0.009	0.001
ABBC	0.158	0.006	0.113	0.002
ABCA	0.014	0.001	0.010	0.001
ABCB	0.070	0.004	0.076	0.002
ABCC	0.048	0.003	0.077	0.002
ABCD	0.189	0.000	0.197	0.000

Table 8: Distribution of rhymes and repetitions in QuaTrain.

Human	Generated	Score
But give me leave , in proper time , To rearrange the broken chime With one who is as good , at least , In all respects , as the deceased .	But give me leave, in proper time, To rearrange the broken chime With one who is as good and sad as he, Till one straight line of light appear,	0.711
Of the tenants of the wood ; How the tortoise bears his shell , How the woodchuck digs his cell , And the ground mole sinks his well	Look! It's red!" How the tortoise bears his shell, How the woodchuck digs his cell, And the ground mole sinks his shell, -	0.836
That lent such splendors to the gypsy brown Of sunburnt cheeks what more could woman do To show her pious zeal ? They went astray , But nature led them as it leads us all .	That lent such splendors to the gypsy brown Of sunburnt cheeks what more could woman do To show her pious zeal? They went astray, But nature led them as it leads us all.	0.98

Table 9: poetry-GPT2-small

Human	Generated	Score
Which one was e'er so dear , so kind , So beautiful as Longing ? The thing we long for , that we are For one transcendent moment ,	Which one was e'er so dear, so kind, So beautiful as Longing? The thing we long for, that we are For one transcendent moment,	0.98
And every shore encircles thine . Rule , Britannia , Britannia rule the waves , Britons never shall be slaves .	I can remember thee. Rule, Britannia, Britannia rule the waves, Britons never shall be slaves.	0.82
In the kingdom of Wabasso , In the land of the White Rabbit . He it was whose hand in Autumn Painted all the trees with scarlet ,	From the courts of Shawondasee, From the land of the White Rabbit. He it was whose hand in Autumn Painted all the trees with scarlet,	0.855

Table 10: poetry-byGPT5-medium

Human	Generated	Score
From persecution of misguided folk , Accustoming my spirit to ignore The burden of the cross , while picturing The bliss of disembodied souls , the grace	To carve so well, upon thy self-built throne Accustoming my spirit to ignore The burden of the cross, while picturing The bliss of disembodied souls, the grace	0.805
Not only high but mean ! With wisdom 's strength He filled our mitred Wilfred , born to rule ; To saintly Cuthbert gave the spirit of prayer ; On me , as one late born , He lays a charge	Not only high but mean! With wisdom's strength He filled our mitred Wilfred, born to rule; To saintly Cuthbert gave the spirit of prayer; On me, as one late born, He lays a charge	0.981

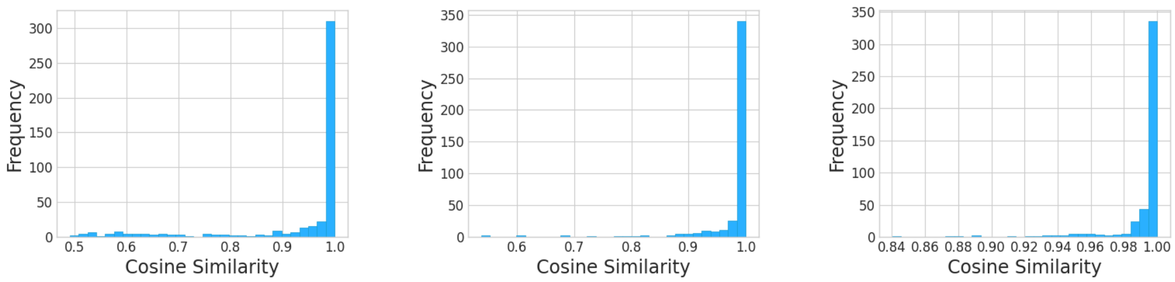
Table 11: poetry-GPTNeo-xl

Method	GPT2-large	p-GPTneo-xl	p-byGPT5-medium
Vanilla$_{t1.0-p1.0-k0}$	0.002	0.89	0.006
Contrastive$_{k10-\alpha0.6}$	0.000	0.96	0.006
Contrastive$_{k6-\alpha0.7}$	0.000	0.98	0.004
Nucleus$_{p0.9-t0.7}$	0.002	0.99	0.010
Nucleus$_{p0.9-t1.0}$	0.004	0.98	0.004
Nucleus$_{p0.7-t0.7}$	0.012	0.99	0.020
Nucleus$_{p0.7-t1.0}$	0.002	0.99	0.010
Top-k$_{k10-t0.7}$	0.000	0.99	0.008
Top-k$_{k10-t1.0}$	0.002	0.98	0.004
Top-k$_{k25-t0.7}$	0.004	0.99	0.012
Top-k$_{k25-t1.0}$	0.004	0.94	0.004

Table 12: Memorization scores $m_{0.7}$ for samples generated by three models using various decoding methods. Vanilla means that no particular decoding strategies have been applied. Contrastive refers to contrastive search.

Method	$m_{0.7}$	mean	std	min	max
Vanilla$_{t1.0-p1.0-k0}$	0.002	0.55	0.06	0.33	0.77
Contrastive$_{k10-\alpha0.6}$	0.000	0.57	0.06	0.37	0.82
Contrastive$_{k6-\alpha0.7}$	0.000	0.58	0.06	0.39	0.72
Nucleus$_{p0.9-t0.7}$	0.002	0.57	0.07	0.38	0.81
Nucleus$_{p0.9-t1.0}$	0.004	0.57	0.06	0.37	0.97
Nucleus$_{p0.7-t0.7}$	**0.012**	**0.59**	0.07	**0.44**	**0.99**
Nucleus$_{p0.7-t1.0}$	0.002	0.56	0.06	0.37	0.91
Top-k$_{k10-t0.7}$	0.000	0.58	0.07	0.33	0.84
Top-k$_{k10-t1.0}$	0.002	0.57	0.07	0.42	0.95
Top-k$_{k25-t0.7}$	0.004	0.57	0.07	0.39	0.98
Top-k$_{k25-t1.0}$	0.004	0.56	0.06	0.34	0.94

Table 13: Memorization vs. semantic similarity for English unconditioned GPT2-large. The highest values (except for std) are displayed in bold.

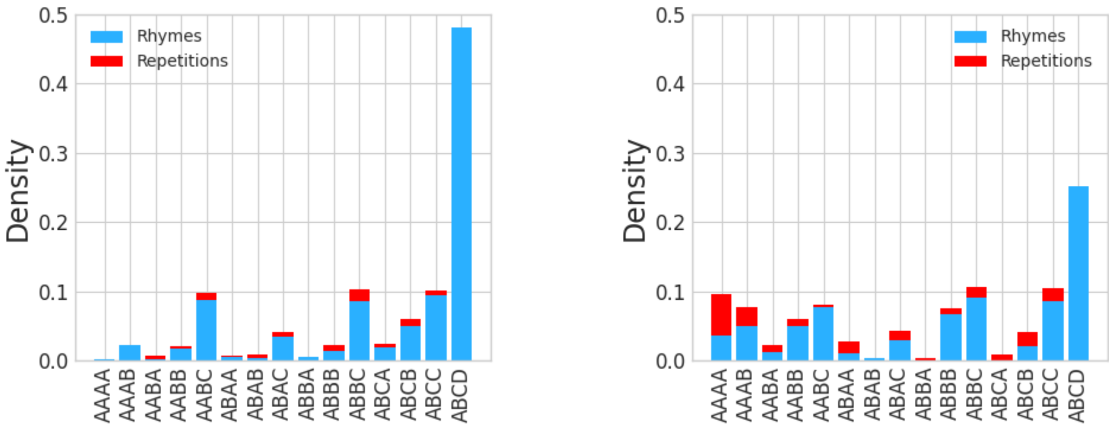


(a) Vanilla$_{t1.0-p1.0-k0}$ (b) Contrastive$_{k6-\alpha0.7}$ (c) Nucleus$_{p0.7-t0.7}$

Figure 9: Semantic similarity plots for samples generated by style-conditioned English GPTneo-xl when different decoding strategies are applied.

Method	mean	std	min	max	h	l_1
Vanilla$_{t1.0-p1.0-k0}$	29.83	13.64	4	150	0.75	5.46
Top-k$_{k25-t1.0}$	37.77	20.58	4	150	0.87	4.48
Nucleus$_{p0.7-t0.7}$	35.67	8.85	13	64	0.86	1.89

Table 14: English unconditioned byGPT5-base: impact of sampling on length. The used measures are the mean length, the standard deviation of length, minimal length, maximal length, histogram intersection h and Wasserstein distance l_1.



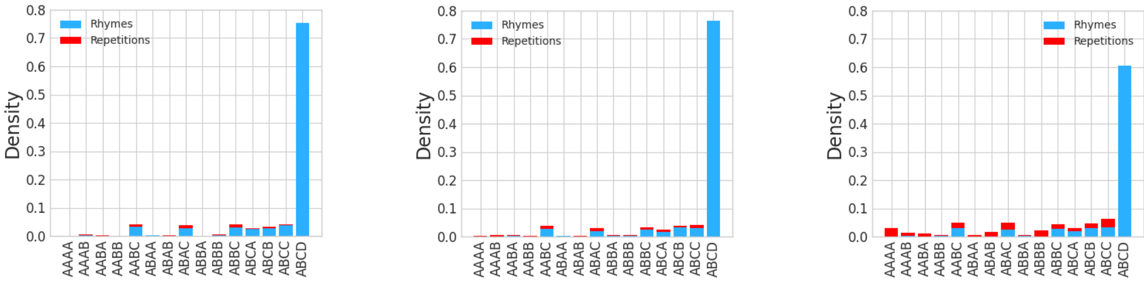
(a) Vanilla$_{t1.0-p1.0-k0}$

(b) Temperature$_{t0.7}$

Figure 10: Rhyme distributions for Structured Adversary: Vanilla vs. lowered temperature.

Model	lang	real rhymes	repetitions
Deepspeare$_{t1.0}$	en	0.72	0.00
Deepspeare$_{t0.7}$	en	0.84	0.00
Structured Adversary$_{t1.0}$	en	0.44	0.08
Structured Adversary$_{t0.7}$	en	0.53	0.21
Deepspeare$_{t1.0}$	de	0.57	0.00
Deepspeare$_{t0.7}$	de	0.66	0.00

Table 15: Real rhymes vs. repetitions: cumulative distributions for Deepspeare and Structured Adversary. The model index denotes the temperature used during inference.



(a) Top-k$_{k25-t1.0}$

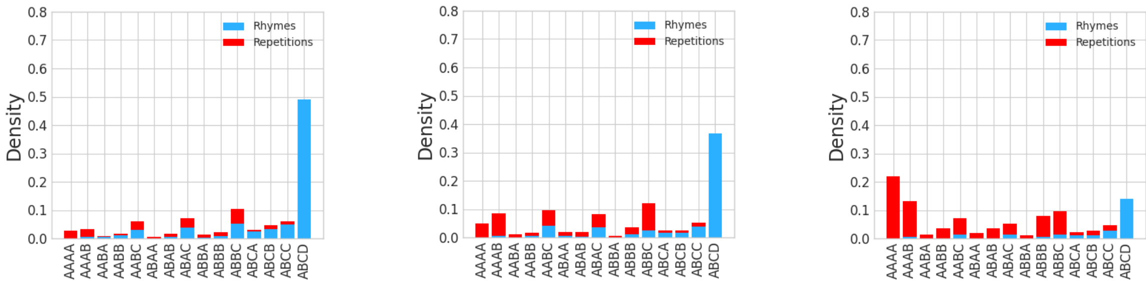
(b) Nucleus$_{p0.7-t1.0}$

(c) Nucleus$_{p0.7-t0.7}$

Figure 11: Distribution of rhyme schemes for samples generated by unconditioned German GPT2-large when different decoding strategies are applied.

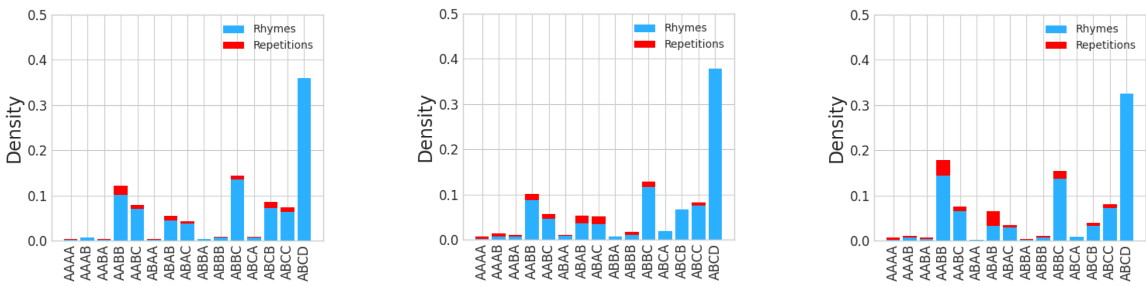
method	real rhymes	repetitions
Vanilla$_{t1.0-p1.0-k0}$	0.156	0.022
Top-k$_{k25-t1.0}$	0.196	0.052
Nucleus$_{p0.7-t1.0}$	0.164	0.071
Nucleus$_{p0.7-t0.7}$	0.176	0.220
Contrastive$_{k10-\alpha0.6}$	0.170	0.103
Contrastive$_{k6-\alpha0.7}$	0.181	0.124
Nucleus$_{p0.9-t1.0}$	0.146	0.030
Nucleus$_{p0.9-t0.7}$	0.168	0.131
Top-k$_{k25-t0.7}$	0.191	0.112
Top-k$_{k10-t1.0}$	0.175	0.153
Top-k$_{k10-t0.7}$	0.194	0.087

Table 16: Real rhymes vs. repetitions: cumulative distributions for unconditioned German GPT2-large. All decoding variants are presented.



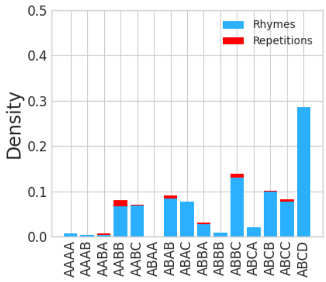
(a) Contrastive$_{k6-\alpha0.7}$ (b) Top-k$_{k10-t0.7}$ (c) Nucleus$_{p0.7-t0.7}$

Figure 12: Distribution of rhyme schemes for samples generated by unconditioned German byGPT5-medium when different decoding strategies are applied.

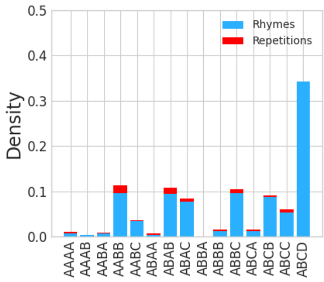


(a) Nucleus$_{p0.7-t1.0}$ (b) Top-k$_{k25-t0.7}$ (c) Nucleus$_{p0.7-t0.7}$

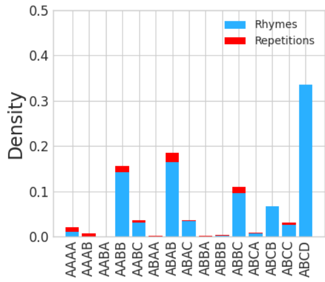
Figure 13: Distribution of rhyme schemes for samples generated by style-conditioned English GPT2-small when different decoding strategies are applied.



(a) Contrastive$_{k10-\alpha0.6}$



(b) Nucleus$_{p0.9-t0.7}$



(c) Nucleus$_{p0.7-t0.7}$

Figure 14: Distribution of rhyme schemes for samples generated by style-conditioned German byGPT5-base when different decoding strategies are applied.

Method	ATTR ↑	MATTR ↑	MTLD ↑
Vanilla$_{t1.0-p1.0-k0}$	**0.804**	**0.794**	**64.679**
Contrastive$_{k10-\alpha0.6}$	0.757	0.744	44.942
Contrastive$_{k6-\alpha0.7}$	0.716	0.717	36.561
Nucleus$_{p0.9-t0.7}$	0.537	0.537	17.297
Nucleus$_{p0.9-t1.0}$	0.720	0.690	33.151
Nucleus$_{p0.7-t0.7}$	0.419	0.461	14.730
Nucleus$_{p0.7-t1.0}$	0.586	0.622	21.757
Top-k$_{k10-t0.7}$	0.611	0.588	21.005
Top-k$_{k10-t1.0}$	0.766	0.754	46.107
Top-k$_{k25-t0.7}$	0.642	0.637	24.289
Top-k$_{k25-t1.0}$	0.797	0.779	58.490

Table 17: Lexical diversity scores for samples generated by English byGPT5-base.