Advances in Science, Technology and Engineering Systems Journal
Vol. 4, No. 6, 115-125 (2019)
www.astesj.com
Special Issue on Multidisciplinary Sciences and Engineering

ASTES Journal
ISSN: 2415-6698

Learning Literary Style End-to-end with Artificial Neural Networks

Ivan P. Yamshchikov*,[1], Alexey Tikhonov[2]

[1]Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany

[2]Yandex, Berlin, Germany

A R T I C L E I N F O

Article history:
Received: 20 June, 2019
Accepted: 26 October, 2019
Online: 28 November, 2019

Keywords:
Stylized text generation
Poetry generation
Artificial neural networks

A B S T R A C T

This paper addresses the generation of stylized texts in a multilingual setup. A long short-term memory (LSTM) language model with extended phonetic and semantic embeddings is shown to capture poetic style when trained end-to-end without any expert knowledge. Phonetics seems to have a comparable contribution to the overall model performance as the information on the target author. The quality of the generated texts is estimated through bilingual evaluation understudy (BLEU), a new cross-entropy based metric, and a survey of human peers. When a style of target author is recognized by the humans, they do not seem to distinguish generated texts and originals.

1 Introduction

This paper is an extension of work presented initially in [1] enhanced with the reasoning and experiments described in [2] and [3].

Generation of authentic, stylistically expressive texts is still challenging despite numerous recent advancement in machine-generation of texts. The ability to generate texts that feel personal and expressive opens numerous possibilities for industrial and scientific applications, see [4] or [5]. For example, the generation of stylized texts could improve user experience in human-machine interfaces. When talking about interactions between a human and a machine, one tends to speak in terms of user experience or interface instead of communication. This is partly due to the fact that predictability in a narrow context was for many years (and still is) one of the main aspects of human-machine interfaces [6], [7]. Communication, on the other hand, naturally implies a certain degree of surprise and context shifts on the way [8]–[10]. We expect understandable and reproducible behavior when interacting with a machine, yet regularly allow other humans to be more versatile and obscure when communicating with us. Yet modern technologies that comply with these expectations tend to be perceived as flat and dull. Indeed, personal assistants such as Alexa or Siri have limited anthropomorphic features. However, they can not be tailored to the needs and preferences of every end-user. Other AI-powered solutions tend to have no anthropomorphism at all, yet humans voluntarily interact with them. One such example would be a game of advanced chess. In advanced chess, two players accompanied by two AIs play against one another. The concept of such a game was proposed in the 1970s [11], but the history of advanced chess started in 1998, just one year after the historic game between Garry Kasparov and Deep Blue. It represented a paradigm shift, from the twentieth-century 'humans against the machines' to the twenty-first-century 'humans with the machines.' Stylistically expressive text generation can facilitate and speed up this shift.

Modern generative models are typically trained on massive corpora of texts. These corpora consist of various texts created by different people in diverse genres and forms. Standard generative models do not incorporate the information on authorship in any manner and therefore learn to generate texts that are some sort of 'average' across the training dataset. Such texts could hardly be perceived as human-written. Indeed, there is an intuitive belief that each human has a recognizable writing style. This intuition, however, is not backed up by any quantitative definition of a literary style. Experts can have an extensive discussion on what distinguishes a text written by Edgar Allan Poe from a text written by Walt Whitman. However, as far as we are concerned, these expert opinions can rarely be quantified in the context of a rigorously defined natural language processing (NLP) task. That is the reason why this paper addresses the problem of author styl-

*Corresponding author: Ivan P. Yamshchikov, Inselstrasse 22, 04103, Leipzig, Germany, ivan@yamshchikov.info ized text generation in an end-to-end setup. We propose a model that generates texts resembling the writing style of a particular author without any external information on stylistic aspects of the author's writing. We also restrict the scope of this paper to poetry rather than prose due to several reasons. First, the same machine learning model trained on a smaller dataset tends to show weaker performance. Naturally, for a given amount of authors, a poetic corpus would be inevitably smaller in terms of the number of available texts than a corpus that includes prose. This makes the problem more challenging. However, one can argue that if a proposed method manages to reproduce style when trained on a relatively small corpus of poems, it would be applicable to a bigger dataset as well. Second, poetry is often regarded as a more stylistically expressive form of literary art. This gives hope that even with relatively short chunks of generated texts, one could easily distinguish the style of a target author. One of the methods to estimate the performance of the generative model is to measure how often humans attribute the generated texts to the target authors. The expressive nature of poetry makes such assessment easier for the peers and, therefore, facilitates the experiments. Both these factors make poetry generation a more challenging and a more exciting task than the generation of prose. Nevertheless, we have every reason to believe that a model that successfully generates stylized poetry would also successfully mimic the style of a prosaic text.

This paper has four significant contributions: (1) formalization of stylized text generation framework; (2) a new sample cross-entropy metric that could be used to measure the quality of the stylization; (3) a long short-term memory artificial neural network with extended phonetic and semantic embeddings that generates stylized texts that are quantified both subjectively through a survey and objectively with sample cross-entropy and BLEU metrics; (4) the proposed approach is applicable to a multilingual setting. Most importantly, we show that modern generative neural networks can learn and mimic the author's style in an end-to-end manner without any external expert knowledge.

2 Related work

Almost a century ago, [12] suggested that computers can generate poetry algorithmically. In [13], one can find a detailed taxonomy of generative poetry techniques. However, to work with stylized poetry generation, one has to resolve the ambiguity of the literary style. In a rapidly developing field of textual style-transfer, one can see various definitions of literary style that often contradict one another [14]. Indeed, different aspects of literary style could includ a sentiment of a text (see [15] or [16]), it's politeness [17] or a so-called style of the time (see [18]). The style of the time aspect is also addressed by [19] and by [20]. A paper by [21] generalizes these ideas and proposes to measure every stylistic aspect with a dedicated external classifier. Yet the problem of style transfer differs from the stylized text generation significantly: a human-written input could be used to control the saliency of the output [22]. This might improve the resulting quality of the texts that are generated. In the meantime, such input is not available for the generative model, and it has to generate stylized texts from scratch. This makes our problem setup, similar to [23]. This paper shows that modern artificial neural networks can learn the literary style of an author end-to-end without any predefined expert knowledge. This is a meaningful empirical result in itself since it serves as an initial proof-of-concept for further research of stylized text generation. In this paper, we focus on RNN-based generative models. Let us briefly mention several contributions in that area that are relevant for further discussion.

There are a few papers that decompose content and semantics of the input, see [24]–[29]. Yet none of them works with the literary style of the generated texts. In [30] the authors developed a persona-based model that was to be consistent in terms of response generation. There are works that specifically address Chinese classical poetry generation, for example [31]–[35]. However, the contributions in the area of generative poetry in languages other than Chinese or in a multilingual setting are relatively rare. In [36], an algorithm generates a poem in line with a topic given by the user, and in [37], the authors generate stylized rap lyrics with LSTM trained on a rap poetry corpus. These contributions are all based on the idea that when trained on a corpus of similar documents, an artificial neural network could infer certain properties of such text and reproduce them in generative mode. We suggest to make one step further and develop a model that could be trained on a broad dataset of various documents yet generate texts that resemble various subsets of the training dataset.

3 Generation of stylized texts

Consider a corpus $C = \{T_i\}_{i=0}^{M}$ of M texts written in a given language. Every text of length l is a sequence $T_i = (w_j)_{j=0}^{l}$ where words (denoted here as w_j) are drawn from a vocabulary set $V = \{w_j\}_{j=1}^{L}$, where L is the size of a given vocabulary. One can try to vary the length of T_i in training. For practical reasons we further assume that T_i are lines of poems. Sine these are the text in natural language there is additional structure over the set V and certain principles according to which sequences T_i are formed. However, this information is only partially observable.

In a standard problem of text generation, a language model predicts the next word w_k using a conditional probability $P(w_k|(w_i)_{i=0}^{k-1})$. Neural networks are widely used for language modeling since [38], see also [39] and [40]. In particular, this family of algorithms allows to avoid the dimensionality curse of a classical language model. One typically would like to obtain a mapping $Y : (C, \mathbb{R}^m, F) \to \mathbb{R}^d$ and then train a model such that $G(C) : \mathbb{R}^d \to \{T_i^G\}$.

Usually one tries to use additional observable information in order to improve the general performance of the model, e.g. [41]. For example, if there is a performance metric D (such as BLEU, F1, etc.), one usually minimizes $D(\{T_i\}, \{T_i^G\})$, where $\{T_i\}$ is a randomized sample of C. Let us find a stylization model $G(C|S)$ that takes into consideration a subset S of continuos and categorial variables out of $(\mathbb{R}^m, F)$ and a metric D in such a way that

$$G(C|S) : \begin{cases} (C, \mathbb{R}^m, F) \to \{T_i^G\} \\ \{T_i^G|S\} \sim \{T_i|S\} \text{ w.r.t. } D \end{cases} \quad (1)$$

A stark difference in our approach is that we train our model using all information available to us, i.e., $(C, \mathbb{R}^m, F)$. However, we do not measure its overall performance, but rather test it on a specific domain S. This core idea is, in some sense, similar to the principles of one-shot learning, see [42], and, generally, transfer learning, see [43], and author-attribution methods, see [44]. The model has access to information about a broader domain of data. This information happens to exogenous to the problem in its' narrow formulation. For a given domain S, it can seem irrelevant, yet it can significantly boost the results of the model. There are several advantages associated with this approach. First of all, such models naturally imply ample opportunities for customization. If one wants some additional control, the parameters of the model one can include them into S. The output $\{T_i^G|S\}$ will resemble original texts $\{T_i|S\}$ that satisfy S conditions. That property makes such methods applicable to personalized interfaces. At the same time, one would expect that due to the 'umbrella' structure in which $G(C|S)$ is trained on the whole corpus $(C, \mathbb{R}^m, F)$ the model would outperform various smaller models trained on different subsamples of C. Artificial neural networks are known to generalize well. This allows to speculate that system he system that uses less information for training would show inferior performance when compared with a system that is trained on the whole corpus C.

Further we work with an artificial neural network that uses the name of an author as such condition S. Such models when trained on a sufficiently large dataset, generates lyrics similar to the text written by a target author. The quality of the stylization is assessed both subjectively (based on a survey of respondents) and objectively (in terms of BLEU and a sample cross-entropy that we define further). The model has been trained with English and Russian datasets. There are no obstacles to implement this model in other languages. Before we describe our model in detail, let us briefly address one of the issues of poetry generation that rarely gets enough attention from researchers, namely, the phonetics of the lyrics.

4 Importance of phonetics

The structure of the poem could be different across different languages. For example, Chinese poems have highly specific and distinct structures, see [35], whereas some American poetry of the twentieth century or so-called white poems in Russian hardly have any definite structure at all. The structure of poems can depend on various factors. These factors are primarily phonetic. In the broadest sense, a sequence of tones in a Chinese quatrain, a structure of a Persian ruba'i, rhymes in a classical western sonnet, or a structure within rap bars could be expressed as a set of phonetic rules based on a particular understanding of expressiveness and euphony shared across a given culture. Some cultures and styles also have particular semantic limitations or 'standards,' for example, 'centrality' of specific topics in classical Japanese poetry, see [45]. We do not make attempts to address high-level semantic structure. However, one can add some pseudo-semantic rules to our model, say via some mechanism in line with [36] or [32]. Instead, we suggest focusing on syntax and phonetics.

The importance of phonetics in poetical texts was broadly discussed among Russian futuristic poets, see [46]. Several Russian linguistic circles and art groups (particularly OPOJAZ) in the first quarter of 20th century were actively discussing the concept of the abstruse language, see [47], also stressing that the form of a poem, and especially its acoustic structure, is a number one priority for the future of literature. In their recent paper, [48] have challenged the broadly accepted idea that sound and meaning are not interdependent: unrelated languages very often use (or avoid) the same sounds for specific referents. [49] backs this idea up, showing that modern word embeddings capture such dependencies to a certain respect. In line with these ideas, one of the critical features of the model that we discuss below is its concatenated embedding that contains information on the phonetics of every word preprocessed by a bi-directional LSTM alongside with its vectorized semantic representation.

5 Model

Let us look at an LSTM-based language model that predicts the w_{n+1} word based on $w_1, ..., w_n$ previous inputs and other macro parameters of the sequence. Typically one passes the needed parameter to the network by writing it in its initial state. However, as the generated sequence gets longer, the network often 'forgets' the parameters of the document. We want to develop a model to address the problem given in (1). To do that, we support our model at every step with the embeddings of the document that is currently being analyzed (these are variables from the set S on which we want to condition our model during the generation process, such as the name of a document or its author). This idea makes our approach different from a classical word-based LSTM. It was used in [50] for stylized music generation. Figure 1 shows a schematic picture of the model; document information projections are highlighted with blue and white arrows.

We used an LSTM with 1152-dimensional input and 512-dimensional projection of a concatenated author and document embeddings. A concatenated word representation is shown schematically in Figure 2. At every step and use the word embeddings, which are generally used for a task of this type. However, we believe that word embeddings are not enough: the char-based information might allow for some useful grammar rules to be learned, and, as we discussed in the previous section, phonetics plays a crucial role in the euphony of the final output. To include this information, we suggest using two bidirectional LSTMs with a 128-dimensional vector as a final state. One of the LSTMs works with a char-representation of the word whereas another uses phonemes of the International Phonetic Alphabet[1], employing heuristics to transcribe words into phonemes. Before training the model, we obtained a vocabulary V for the training dataset, then applied transcribing heuristics to it. The result was a phoneme-based vocabulary V_p. Then, in the training phase, the char-based bidirectional LSTM was reading words from the initial vocabulary $w_j \in V$, and the phoneme-based LSTM was reading words from the heuristically transcribed vocabulary $w_j^{'} \in V_p$. [51] proposes a somewhat similar idea, but with convolutional neural networks rather than with LSTMs. The bidirectional LSTM approach is new to our knowledge.

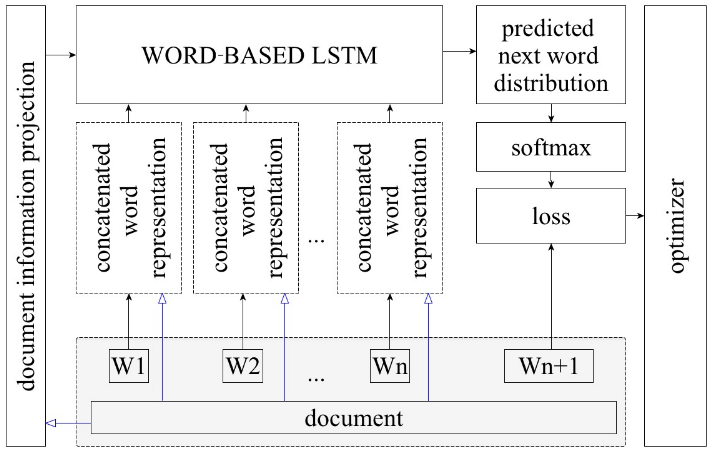
document information projection
WORD-BASED LSTM
predicted next word distribution
softmax
loss
optimizer
concatenated word representation
concatenated word representation
...
concatenated word representation
W1
W2
...
Wn
Wn+1
document

Figure 1: The scheme of the language model. Document information projections are highlighted with blue and white arrows. The projections on a state space of the corresponding dimension is achieved with simple matrix multiplication of document embeddings.

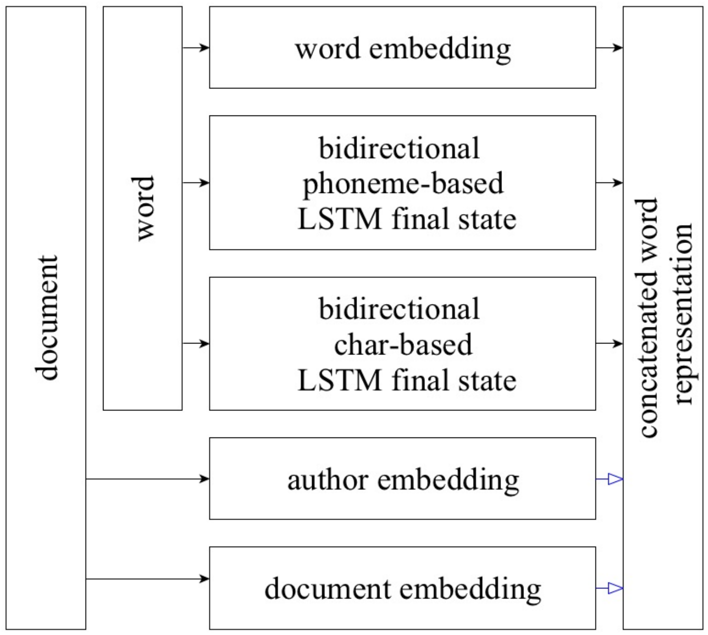
document
word
word embedding
bidirectional phoneme-based LSTM final state
bidirectional char-based LSTM final state
author embedding
document embedding
concatenated word representation

Figure 2: Concatenated word representation.

[1]https://en.wikipedia.org/wiki/International_Phonetic_Alphabet

In Section 7, we describe a series of objective and subjective tests that we ran across a generated output $\{T_i|S\}$, but first, let us briefly describe the datasets used for training.

6 Datasets

Two proprietary datasets of English and Russian poetry were used for training. Every character was transferred to a lower case, and punctuation was deleted. No other preprocessing was made. The overview of the sizes of the datasets can be found in Table 1. This allowed having approximately 330 000 verses in train dataset and another 10 000 verses forming a test dataset for Russian poetry. For English poetry train data consisted of 360 000 verses with approximately 40 000 verses forming the test data.

Table 1: Parameters of the training datasets.

	N. of documents	Size of vocab.	N. of authors	Size
English	110000	165000	19000	150 Mb
Russian	330000	400000	1700	140 Mb

Table 2: Number of words in the training datasets for human-peer tested lyrics.

	N. of words		N. of words
Shakespeare	10 218	Pushkin	226 001
Carroll	19 632	Esenin	73 070
Marley	22 504	Letov	29 766
MUSE	7 031	Zemfira	23 099

During the training phase, the beginning and end of every text T_i were tokenized. In the generation phase, the network is conditioned on values of document parameters S and is initialized with a special 'start' token. As we stated previously, in the most general setup, the set of variables S on which we want to condition our generative model $G(S)$ can vary. Here we use only one categorical variable - the name of the author to test the proposed mechanism for the stylized text generation. We trained the model for Russian (Alexander Pushkin, Sergey Esenin, Joseph Brodsky, Egor Letov and Zemfira Ramazanova) and for English (running tests on lyrics of William Shakespeare, Edgar Allan Poe, Oscar Wilde, Lewis Carroll and Bob Marley as well as lyrics of the American band Nirvana and UK band Muse). Table 2 shows that there were far more authors in the dataset. For the experiments, we chose more prominent authors who are known for their poetic styles. A fluent speaker of the target language could easier identify these authors. The model produces results of comparable quality for both languages, so to make this paper shorter, we further address generative poems in English only and provide the experimental results for Russian in the Appendix. We do not see any difficulties that could prevent the implementation of the proposed model in other languages. As long as there is a sufficiently large corpus C and a phonetically transcribed vocabulary V_p, one could use such a method.

The model produces results of comparable quality for both languages, so to make this paper shorter, we further address generative poems in English only and provide the experimental results for Russian in the Appendix.

Table 3 shows some examples of the generated stylized poetry. The model seems to capture the syntactic characteristics of the author (look at the double negation in the Neuro-Marley lyrics, for example). The vocabulary that is characteristic for a target author alsoo seems to be captured. 'Darkness,' 'burden,' 'fears' could be subjectively associated with the gothic lyrics of Poe. At the same time, 'fun,' 'sunshine,' 'fighting every rule' could be associated with positive yet rebellious reggae music. Author-specific vocabulary can technically imply specific phonetics that characterizes a given author; however, this implication is not self-evident, and generally speaking, does not have to hold. As we demonstrate later, phonetics does, indeed, contribute to the author's stylization significantly.

7 Experiments and evaluation

There is a variety of metrics that could be used as D, for which we could estimate the quality of the generated texts. A standard approach for a comparison of two generative models is to measure cross-entropy loss at certain checkpoints. However, as [52] writes: "There can be significant differences in final performance across checkpoints with similar validation losses." In the case of the provided model, a cross-entropy does not seem to give any meaningful information. To quantitatively estimate the final model $G(C|S)$, we carried out an ablation experiment. We used a plain vanilla LSTM without word-by-word document information support that was only using classic word embeddings as a baseline. Another model to compare with included document information support but didn't have bidirectional LSTMs for phonemes and characters. All the models have shown comparable values of cross-entropy loss after an equal amount of epochs. It seems that the additional structures are not facilitating learning but are also not hindering it. Below we describe two automated metrics that are natural for the problems of such type, namely, the BLEU and the sample cross entropy. We demonstrate that the model captures styles of different authors and that instead of developing several different models $\{G(S_i)\}$ trained on corpora corresponding to different authors one can use one model $G|S$ and change S depending on the style that one wants to reproduce.

7.1 Sample cross entropy

Cross entropy seems to be one of the most natural theoretic-informational metrics to estimate the similarity of different discrete sequences. To distinguish it from the cross-entropy loss, we further call it the sample cross entropy. We calculate it as follows. One samples several subsets with the same length (in words) from the original author texts. The sample is drawn so that there is a comparable number of unique texts for each author within the sample. Then one splits the texts of a given author A_i into two random groups and calculates the pairwise[2]cross-entropy between original texts of the author A_i and texts generated by the model $\{T_i^G|A_i\}$. The cross-entropy between the obtained sets of texts was calculated with MITML, see [53], as follows. For every sample written by the author, one built a standard 3-gram based language model with standard MITML smoothing. A shared vocabulary across all samples was calculated. Then the perplexity was computed by applying the language models based on the author-written texts to the original and generated texts. Though both values have a similar meaning, we suggest applying logarithm to the obtained value to get the cross-entropy instead of the perplexity. In Table 4 one can see the results of these calculations. Analogous results for Russian can be found in Appendix in Table 10. The model clearly captures the style of the time mentioned earlier alongside the individual styles. Generated texts stylized for the authors from a similar time period tend to demonstrate lower sample cross-entropy with human-written texts written close to that time.

The model captures the author's writing style better if the sample cross-entropy between the texts written by the given author and the texts generated by the model is lower. We also provide cross-entropy between random samples from the texts of the same author to demonstrate how self-similar are the human-written texts. The majority of texts in the training dataset were the texts from the 20th century. One can see that the model 'perceives' texts of Edgar Allan Poe or William Shakespeare to be closer to the lyrics of Oscar Wilde and Lewis Carrol than to the samples of the original texts. At the same time, Poe and Shakespeare are also reasonably well approximated by the model. To give a reference, we also provide cross-entropies between the texts sampled randomly out of the vocabulary as well as the texts obtained through a weighted average sampling and the human-written texts.

7.2 BLEU

BLEU is a metric estimating the correspondence between a human-written text and a machine's output. It is typically used to assess the quality of of machine translation. We suggest to adopt it for the task of stylized text generation in the following way: a random starting line out of the human-written poems is used to initialize the generation. Generative model 'finishes' the poem generating thee ending lines of the quatrain. Then one calculates BLEU between lines generated by the model and three actual lines that finished the human-written quatrain.

First of all, one can see that it makes sense to train the model on the whole dataset and then restrict it to the chosen author rather than train it on the target author exclusively. This is shown in Table 5.

[2]Hence 'sample' in the name of the metric.

Table 3: Examples of the generated stylized quatrains. The punctuation is omitted since it was omitted in the training dataset.

Neuro-Poe	Neuro-Marley
her beautiful eyes were bright	don t you know you ain t no fool
this day is a burden of tears	you r gonna make some fun
the darkness of the night	but she s fighting every rule
our dreams of hope and fears	ain t no sunshine when she s gone

Table 4: Sample cross entropy between generated texts $\{T_i^G|A_i\}$ and actual texts for different authors. The two smallest values in each row are marked with * and ** and a bold typeface. The sample cross entropy between random samples from the texts of the target author and randomly generated sequences of words (uniform and weighted respectively) as well as other samples written by the same author (denoted as SELF) are shown for reference.

Model $G(A_i)$/ author	Shakespeare	Poe	Carroll	Wilde	Marley	Nirvana	MUSE
Neuro-Shakespeare	**19.0****	21.6	**18.5***	19.9	21.8	22.0	22.4
Neuro-Poe	22.0	**20.4****	21.2	**19.0***	26.0	25.4	26.0
Neuro-Carroll	22.2	23.6	**18.9***	22.5	22.4	**21.8****	23.8
Neuro-Wilde	21.2	20.9	**20.5****	**18.4***	24.5	24.8	26.4
Neuro-Marley	24.1	26.5	22.0	27.0	**15.5***	**15.7****	16.0
Neuro-Nirvana	23.7	26.2	20.0	26.6	19.3	**18.3***	**19.1****
Neuro-MUSE	21.1	23.9	18.5	23.4	17.4	**16.0****	**14.6***
Uniform Random	103.1	103.0	103.0	103.0	103.5	103.3	103.6
Weighted Random	68.6	68.8	67.4	68.5	68.5	68.0	68.0
SELF	23.4	21.8	25.1	27.3	20.8	17.8	13.3

Table 5: BLEU for the full model trained on one particular author dataset, $G(S)$, and on the whole dataset, $G(C|S)$, calculated on the chosen author validation dataset and on the validation dataset that includes a variety of authors. The results may vary across authors depending on the relative sizes of S and C but the general picture does not change.

Model $G(A_i)$	Chosen author S	Validation dataset
$G(S)$	33.0%	19.0%
$G(C|S)$	37.3%(+13%)	37.6%(+98%)

The model $G(S)$ is trained on texts of a particular domain S. In this case the domain corresponds to a particular author. One can see that $G(C|S)$ when tested on the lyrics of the chosen author outperforms $G(S)$. Moreover, $G(C|S)$ performs almost twice as good as $G(S)$ on the validation dataset comprised out of the texts of other authors.

Table 6 summarizes the results of the ablation experiments. One can see BLEU calculated on the validation dataset for the plain vanilla LSTM, LSTM extended with author information but without bidirectional LSTMs for phonemes and characters, and the full model. The uniform random and weighted random give baselines to compare the model.

Table 6: BLEU for uniform and weighted random random sampling, vanilla LSTM, LSTM with author embeddings but without phonetics, and for the full model. Phonetics is estimated to be almost as important for the task of stylization as the information on the target author.

Model $G(A_i)$	BLEU
Uniform Random	0.35%
Weighted Random	24.7%
Vanilla LSTM	29.0%
Author LSTM	29.3% (+1% to vanilla LSTM)
Full model	29.5% (+1.7% to vanilla LSTM)

Table 6 shows that extended phonetic embeddings play a significant role in the overall quality of the generated stylized output. It is essential to mention that phonetics is an inherent characteristic of an author and the training dataset. Humans do not have qualitative insights into phonetics of Wilde or Cobain, yet the information on it turns out to be necessary for the style attribution.

7.3 Survey data

For a survey we sampled quatrains from William Shakespeare, Lewis Carroll, Bob Marley, and MUSE band. Each sampled quatrain was accompanied by a quatrain generated by the model conditioned on the same author. One hundred and forty fluent English-speakers were asked to read all sixteen quatrains in a randomized order and choose one option out of five offered for each quatrain. They had to decide if the author of a given verse was William Shakespeare, Lewis Carroll, Bob Marley, MUSE, or an Artificial Neural Network. Table 7 shows the summary of the obtained results. For analogous results in Russian see in Table 11. A more detailed description of the methodology could be found in the Appendix.The generated texts were human-filtered for mistakes, such as demonstrated in Table 8. This stands to reason since clear mistakes would inevitably allow human peers to detect a generated text. The automated metrics provided above were estimated on the whole sample of generated texts without any human-filtering.

Table 7 shows that the model manages to stylize several given authors. The participants recognized Shakespeare more than 46% of the time (almost 2.5 times more often than compared with a random choice). They did slightly worse in recognizing Bob Marley (40% of cases) and MUSE (39% of

Table 7: Results of a survey with 140 respondents. Shares of each out of 5 different answers given by people when reading an exempt of a poetic text by the author stated in the first column. The two biggest values in each row are marked with * and ** and a bold typeface.

Model $G(A_i)$ or author	Shakespeare	Carroll	Marley	MUSE	LSTM
Neuro-Shakespeare	**0.37***	0.04	0.05	0.14	**0.3****
Shakespeare	**0.46***	0.05	0.04	0.07	**0.3****
Neuro-Carroll	0.02	0.07	**0.26****	0.18	**0.41***
Carroll	0.05	**0.2****	0.14	0.11	**0.32***
Neuro-Marley	0.02	0.01	**0.47***	0.2	**0.29****
Marley	0.15	0.05	**0.4***	0.1	**0.24****
Neuro-MUSE	0.09	0	0.12	**0.34****	**0.39***
MUSE	0.03	0.05	**0.28****	**0.39***	0.2

cases, 2 times higher than a random choice). It seems that the human-written quatrains were recognizable. The participants were also fluent enough in the target language to attribute given texts to the correct author. However, people believed that the generated text was actually written by a target author in 37% of cases for Neuro-Shakespeare, 47% for Neuro-Marley, and 34% for Neuro-MUSE, respectively. Lewis Carroll was less recognizable. His texts were only recognized in 20% of cases. The underperformance of the model can be explained with difficulty experienced by the participants in determining Carroll's authorship. If humans have a hard time recognizing the author's style, it is hard to expect an excellent performance of a model on such a target author. Combining the results in Table 6 with the results of the survey shown in Table 7, one could conclude that the phonetic structure of lyrics has an impact on the correct author attribution of the stylized content. This impact is usually not acknowledged by a human reader but is highlighted with the proposed experiment.

7.4 Discussion

Here we would like to share some of the insights into the nature of computational creativity. First, the stylized generation is thought-provoking since the model learns similarities between various authors in the dataset and extrapolates if there is a deficit of information. For example, one of the Nirvana-stylized lines ran:

a god who's always welcome to iraq

Kurt Cobain committed suicide long before the start of the Iraq campaign. However, since the model was trained on a massive corpus of poems (and not only on Nirvana ones), this line emerged among the generated lyrics. It also fits into the broader context of a song due to the correct rhythmic structure. When filtering the network's output, we found this line exceptionally interesting. It was meaningful to us because it resonated with our intuition that, if he were alive, Cobain would have spoken about the Iraq war in his lyrics. We also found it stylistically similar to Cobain lyrics and emotionally provocative. Formally speaking, all of these characteristics are human-attributed, and yet they can emerge out of explorative human-algorithm interactions. One could reason that the topic of Iraq occurred in other songs by grange bans, and the model extrapolated the saliency of the topic to the Cobain lyrics.

Second, creativity is serendipitous and highly depends on the perception of the viewer. In the Appendix of this paper, we provide examples of typical mistakes. One such mistake that is characteristic for the state of the art artificial neural network is an abrupt end of a line. As we listed the examples of such behavior, we came across the following line:

at night i lay waiting for a

However, one of our colleagues pointed out that this line represents a perfect one-line poem; an abrupt ending only enhances the feeling of longing. We have decided to submit this poem among others into the journal of modern poetry, and it was published[3]. We find both these insights meaningful for further research on computational creativity.

Over the last several years, we have seen several attempts at employing AI as a tool for creators. Using the approach proposed in [54], some visual artists applied style transfer algorithms to various pictures, creating new and unique experiences. In a postmodern cultural context, style transfer can be a conceptual tool that is able to convey the thought of an artist in a more expressive and meaningful way, for example, see [55].

8 Conclusion

This paper addresses the problem of author-stylized text generation. We show that an extension of a language model supported by the document meta-information at every step is apt for such tasks. Large concatenated embeddings that consist of word embedding, a phoneme-based bidirectional LSTM final state, and a char-based bidirectional LSTM final state are shown to facilitate end-to-end learning of author poetic style. Moreover, extending word embeddings with phonetic information has a comparable impact on the BLEU of the generative model as the information on the authors of the text. A cross-entropy method to estimate the quality of stylization is proposed. This model is shown to perform in Russian and English, and there seem to be no obvious reasons for which one could not implement it in other languages. It was shown

[3]https://nokturno.fi/poem/on-author-stylized-neural-network-poetry

that the texts generated by the model tend to be closer to the texts of the target author than the text generated by a plain vanilla LSTM both in terms of the cross-sample entropy and BLEU. It was also shown that, when faced with an author with a recognizable style (an author who is recognized approximately two times more frequently than at random), humans mistakenly recognize the output of the proposed generative model for the target author as often as they correctly attribute original texts to the author in question.

References

[1] Alexey Tikhonov and Ivan P. Yamshchikov. Guess who? multilingual approach for the automated generation of author-stylized poetry. In IEEE Spoken Language Technology Workshop (SLT), pages 787–794, 2018.

[2] Alexey Tikhonov and Ivan P. Yamshchikov. Sounds wilde. phonetically extended embeddings for author-stylized poetry generation. In Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 117–124), 2018.

[3] Ivan P. Yamshchikov and Alexey Tikhonov. I feel you: What makes algorithmic experience personal? In Proceedings of EVA Copenhagen 2018 - Politics of the Machines - Art and After, 2018.

[4] Daniel Livingstone. Turing's test and believable ai in games. Computers in Entertainment, 4(1):6, 2006.

[5] Alan Dix. Human-like computing and human—computer interaction. In Proceedings of the 30th International BCS Human Computer Interaction Conference: Fusion!, page 52, 2016.

[6] J. Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1):36–45, 1966.

[7] S. K. Card, T.P Moran, and A. Newell. The psychology of human-computer interaction. CRC Press, 1983.

[8] Y. Ohsawa. Modeling the process of chance discovery. Chance discovery, pages 2–15, 2003.

[9] W. R. Saab, N. abd Joolingen and B. H. Hout-Wolters. Eliza—a computer program for the study of natural language communication between man and machine. British Journal of Educational Psychology, 75(4):603–621, 2005.

[10] A. Abe. Curation and communication in chance discovery. In Proceedings of the 6th International Workshop on Chance Discovery (IWCD6) in IJCAI, 2011.

[11] D. Michie. Programmer's gambit. New Scientist, 17th of August:329–332, 1972.

[12] Jon Wheatley. The computer as poet. Journal of Mathematics and the Arts, 72(1):105, 1965.

[13] Carolyn Lamb, G. Brown, Daniel, and L. Clarke, Charles. A taxonomy of generative poetry techniques. Journal of Mathematics and the Arts, 11(3):159–179, 2017.

[14] Alexey Tikhonov and P. Yamshchikov, Ivan. What is wrong with style transfer for texts? In arXiv preprint, 2018.

[15] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. 31st Conference on Neural Information Processing Systems, pages 6833–6844, 2017.

[16] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, volume 1, pages 1865–1874, 2018.

[17] Rico Sennrich, Barry Haddow, and Alexandra Birch. Controlling politeness in neural machine translation via side constraints. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 35–40, 2016.

[18] James M. Hughes, Nicholas J. Foti, David C. Krakauer, and Daniel N. Rockmore. Quantitative patterns of stylistic influence in the evolution of literature. Proceedings of the National Academy of Sciences, 109(20):7682–7686, 2012.

[19] Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. Shakespearizing modern language using copy-enriched sequence-to-sequence models. In Proceedings of the Workshop on Stylistic Variation, pages 10 – 19, 2017.

[20] Keith Carlson, Allen Riddell, and Daniel Rockmore. Zero-shot style transfer in text using recurrent neural networks. In arXiv preprint, 2017.

[21] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. In arXiv preprint, 2017.

[22] Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. In arXiv preprint:1709.08878, 2017.

[23] Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. In Proceedings of the Workshop on Stylistic Variation, pages 94 – 104, 2017.

[24] Remi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1203–1213, 2016.

[25] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. In arXiv preprint, 2017.

[26] Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. Context-aware natural language generation with recurrent neural networks. In arXiv preprint, 2016.

[27] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. In International Conference on Machine Learning, pages 1587–1596, 2017.

[28] Alexey Tikhonov, Viacheslav Shibaev, Aleksander Nagaev, Aigul Nugmanova, and Ivan P. Yamshchikov. Style transfer for texts: Retrain, report errors, compare with rewrites. In arXiv preprint:1908.06809, 2019.

[29] Ivan P Yamshchikov, Viacheslav Shibaev, Aleksander Nagaev, Jürgen Jost, and Alexey Tikhonov. Decomposing textual information for style transfer. arXiv preprint arXiv:1909.12928, 2019.

[30] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. A persona-based neural conversation model. CoRR, abs/1603.06155, 2016.

[31] Jing He, Ming Zhou, and Long Jiang. Generating chinese classical poems with statistical machine translation models. In AAAI, 2012.

[32] Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. Generating chinese classical poems with rnn encoder-decoder. In Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pages 211–223, 2017.

[33] Rui Yan. i, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), pages 2238–2244, 2016.

[34] Rui Yan, Cheng-Te Li, Xiaohua Hu, and Ming Zhang. Chinese couplet generation with neural network structures. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 2347 – 2357, 2016.

[35] Jiyuan Zhang, Yang Feng, Dong Wang, Yang Wang, Andrew Abel, Shiyue Zhang, and Andi Zhang. Flexible and creative chinese poetry generation using neural memory. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, volume 1, pages 1364–1373, 2017.

[36] Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. Generating topical poetry. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1183–1191. Association for Computational Linguistics, 2016.

[37] Peter Potash, Alexey Romanov, and Anna Rumshisky. Ghostwriter: Using an lstm for automatic rap lyric generation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1919–1924. Association for Computational Linguistics, 2015.

[38] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. Journal of machine learning research, 3(Feb):1137–1155, 2003.

[39] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. Aistats, pages 246–252, 2005.

[40] Andriy Mnih and Geoffrey Hinton. A scalable hierarchical distributed language model. In Advances in neural information processing systems, pages 1081–1088, 2009.

[41] Yangyang Shi. Language Models with Meta-information. PhD thesis, Technical University of Delft, 2014.

[42] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. In IEEE transactions on pattern analysis and machine intelligence, pages 594 – 611, 2006.

[43] Sebastian Thrun and Lorien Pratt. Learning to learn. Springer Science & Business Media, 2012.

[44] Douglas Bagnall. Author identification using multi-headed recurrent neural networks. In arXiv preprint, 2015.

[45] Senko K. Maynard. The centrality of thematic relations in japanese text. Functions of language, 1(2):229–260, 1994.

[46] Aleksei Kruchenykh. Phonetics of theater. M.:41, Moscow, 1923.

[47] Victor Shklovsky. Poetics: on the theory of poetic language. 18 State typography, Petrograd, 1919.

[48] E Blasi, Damián, Søren Wichmann, Harald Hammarström, F. Stadler, Peter, and H. Christiansen, Morten. Sound – meaning association biases evidenced across thousands of languages. Proceedings of the National Academy of Sciences, 113(39):10818 – 10823, 2016.

[49] Ivan Yamshchikov, Viacheslav Shibaev, and Alexey Tikhonov. Dyr bul shchyl. proxying sound symbolism with word embeddings. In Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP, pages 90–94, 2019.

[50] Alexey Tikhonov and P. Yamshchikov, Ivan. Music generation with variational recurrent autoencoder supported by history. In arXiv preprint, 2017.

[51] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. In arXiv preprint, 2016.

[52] Ziang Xie. Neural text generation: A practical guide. In arXiv preprint, 2017.

[53] Bo-June Hsu and James Glass. Iterative language model estimation: efficient data structure & algorithms. In Ninth Annual Conference of the International Speech Communication Association, 2008.

[54] L.A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2414 – 2423, 2016.

[55] B. Joshi, K. Stewart, and D. Shapiro. Bringing impressionism to life with neural style transfer in come swim. In arXiv preprint:1701.04928, 2018.

A Examples of output

Table 8 lists some illustrative mistakes of the model both for English and for the Russian language. Reading the raw output, we could see several types of recurring characteristic errors that are typical for LSTM-based text generation. They can be broadly classified into several different types:

- If the target author is underrepresented in the training dataset, the model tends to make more mistakes, mostly, syntactic ones;
- Since generation is done in a word-by-word manner, the model can deviate when sampling a low-frequency word;
- Pronouns tend to cluster together, possibly due to the problem of anaphoras in the training dataset;
- The line can end abruptly; this problem also seems to occur more frequently for the authors that are underrepresented in the training dataset.

Table 9 lists some subjectively cherry-picked, especially successful examples of the system outputs both for English and for the Russian language. Since the text is generated line by line, and verses are obtained through random rhyme or rhythm filtering, several types of serendipitous events occur. They can be broadly classified into four different types:

- Wording of the verse that fits into the style of the target author;
- Pseudo-plot that is perceived by the reader due to a coincidental cross-reference between two lines;
- Pseudo-metaphor that is perceived by the reader due to a coincidental cross-reference between two lines;
- Sentiment and emotional ambiance that corresponds to the subjective perception of the target author.

B Cross-entropy for texts in Russian

The cross-entropy between generated texts and samples of human-written texts was calculated as described in Section 7.1. The results are shown in Table 10

One can see that the model achieves the lowest level of cross-entropy when tested on the texts of the target author for all four authors.

C Survey design

The surveys were designed identically for English and Russian languages. We have recruited the respondents via social media; the only prerequisite was fluency in the target language. Respondents were asked to determine authorship for 16 different 4-line verses. The verses for human-written text were chosen randomly out of the data for the given author. The generated verses were obtained through line-by-line automated rhyme and rhythm heuristic filtering. Since LSTMs are not perfect in text generation and tend to have clear problems illustrated in Table 8, we additionally filtered generative texts leaving the verses that do not contain obvious mistakes described above. Each of the 16 questions consisted of a text (in lower case with stripped-off punctuation) and multiple-choice options listing five authors, namely, four human authors and an artificial neural network. Respondents were informed that they are to distinguish human- and machine-written texts. The correct answers were not shown to the respondents. Table 7 shows the results of the survey for English texts and Table 11 for Russian ones. Higher values in every row correspond to the options that were more popular among the respondents when they were presented with the text written by the author listed in the first column of the table.

Table 8: Examples of several recurring types of mistakes that occur within generated lyrics.

Problem	English	Russain
	$Neuro-MUSE$:	$Neuro-Zemfira$:
broken	every step inside on our faces	ты слышишь я слышу
syntax	that i would stop my self going crazy	шаги мои в душу дрожи
'rare' word	$Neuro-Shakespeare$:	$Neuro-Letov$:
brakes	o ho de profundis she says i am on her	иду гляжу в окно гляжу
the line		мне вслед на небо
	$Neuro-Shakespeare$:	$Neuro-Zemfira$:
pronouns	thou here shalt be and thine	тебе ли ты ль за
'entangle'	who will have to my grave	
sentences	$Neuro-Muse$:	$Neuro-Brodsky$:
don't end	at night i lay waiting for a	двух четырех десять лет за углом
	$Neuro-Shakespeare$:	$Neuro-Lenov$:
nonsense	do many a fair honour best of make or lose	о о о о и о

Table 9: Cherry-picked examples of generated lyrics after either rhyme or rhythm filtering illustrating typical serendipities.

Serendipity	English	Russain
	$Neuro-Shakespeare$:	$Neuro-Pushkin$:
peculiar	a sense i may not comprehend	во славу вакха или тьмы
wording	of whom i had not to defend	мы гордо пировали
	$Neuro-Marley$:	$Neuro-Esenin$:
apophenic	oh lord i know how long i d burn	ты под солнцем стоишь и в порфире
plot	take it and push it it s your turn	как в шелку беззаботно горишь
	$Neuro-Carroll$:	$Neuro-Zemfira$:
apophenic	your laugh is bright with eyes that gleam	ветер в голове
metaphor	that might have seen a sudden dream	с красной тенью шепчется
	$Neuro-Muse$:	$Neuro-Letov$:
peculiar	i am the man of this universe	только в ушах отражается даль
sentiment	i remember i still am a curse	только белая смерть превращается в ад

Table 10: Sample cross entropy between generated texts $\{T_i^G|A_i\}$ and actual texts for different authors. The two smallest values in each row are marked with * and ** and a bold typeface. The sample cross entropy between random samples from the texts of the target author and randomly generated sequences of words (uniform and weighted respectively) as well as other samples written by the same author (denoted as SELF) are shown for reference.

Model $G(A_i)$/ author	Pushkin	Esenin	Brodsky	Letov	Zemfira
Neuro-Pushkin	**17.9***	**21.8****	23.4	27.0	30.8
Neuro-Esenin	**20.4****	**18.8***	21.0	22.7	26.0
Neuro-Brodsky	23.5	**21.1****	**17.2***	20.9	23.8
Neuro-Letov	22.2	**20.0****	20.8	**19.6***	23.6
Neuro-Zemfira	19.5	**17.1****	18.1	18.2	**16.6***
Uniform Random	103.0	103.1	103.0	103.0	103.8
Weighted Random	40.8	40.2	40.2	42.6	45.6
SELF	35.0	33.7	38.0	28.3	12.0

Table 11: Results of a survey with 178 respondents. Shares of each out of 5 different answers given by people when reading an exempt of a poetic text by the author stated in the first column. The two biggest values in each row are marked with * and ** and a bold typeface.

Model $G(A_i)$ or author	Pushkin	Esenin	Letov	Zemfira	LSTM
Neuro-Pushkin	**0.31****	0.22	0.02	0.0	**0.44***
Pushkin	**0.62***	0.11	0.03	0.01	**0.23****
Neuro-Esenin	0.02	**0.61***	0.08	0.0	**0.29****
Esenin	0.06	**0.56***	0.07	0.02	**0.29****
Neuro-Letov	0.0	0.02	**0.40****	0.08	**0.51***
Letov	0.0	0.01	**0.61***	0.02	**0.35****
Neuro-Zemfira	0.0	0.06	0.13	**0.4****	**0.41***
Zemfira	0.0	0.02	0.08	**0.58***	**0.31****