Towards Measuring Coherence in Poem Generation

by

Peyman Mohseni Kiasari

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Large language models (LLM) based on transformer architecture and trained on massive corpora have gained prominence as text-generative models in the past few years. Even though large language models are very adept at memorizing and generating long sequences of text, their ability to generate truly novel and creative texts including poetry lines is limited. On the other hand, past research has shown that variational autoencoders (VAE) can generate original poetic lines adhering to the stylistic characteristics of the training corpus. Originality and stylistic adherence of lines generated by VAEs can be partially attributed to the fact that, firstly, VAEs can be successfully trained on small highly curated corpora in a given style, and secondly, VAEs with a recurrent neural network architecture has a relatively low memorization capacity compared to transformer networks, which leads to the generation of more creative texts. VAEs, however, are limited to producing short sentence-level texts due to fewer trainable parameters, compared to LLMs. As a result, VAEs can only generate independent poetic lines, rather than complete and coherent poems. In this thesis, we propose a new model of coherence scoring that allows the system to rank independent lines generated by a VAE and construct a coherent poem. The scoring model is based on BERT, fine-tuned as a coherence evaluator. We propose a novel training schedule for fine-tuning BERT, during which we show the system different types of lines as negative examples: lines sampled from the same vs. different poems. The results of the human evaluation show that participants perceive poems constructed by this method to be more coherent than randomly sampled lines.

Acknowledgements

This thesis would not have been achievable without the support of many people throughout my years as a graduate student at the University of Waterloo.
I want to start by thanking my supervisor Dr. Olga Vechtomova, who guided me through all of my master's. When I contacted Dr. Vechtomova two years ago as a bachelor's student, I hoped to have a chance to continue my academic studies on NLP. She gave me the chance with her valuable guidance over these two years. Her interest in music, poems, and art led me to find poem generation an exciting area to research. I finally chose to work on poem generation coherency with her direction. Without her, this wouldn't have been possible.
And I want to thank my supervisor and the University of Waterloo for funding me.

I want to thank my research group, Utsav Tushar Das, Gaurav Sahu, Brian Zimmerman, and Olivier Poulin, for the meetings and conversations that helped me learn more and for all of the theses I read from them to build my idea.
I want to thank my family, who supported me all my life. A family in Iran missed their son, who is studying abroad. I appreciate their love, support, and patience. And I want to thank my sister, whom I love so much, and she is far away from his brother.

Finally, I'm speechless when it comes to acknowledging how much love, devotion, support, and aliveness I received from you my love, Rana. The distance between my problems to their solution was always a phone call to you. I can't express how much you cared about my academic career and helped me with your kindness, dedication, and knowledge. I can't picture how painful it would have been to live and compose this thesis without you. I love you so much and there is no way that I can give back what you gave to me.

Dedication

My thesis is dedicated to all the current dramas taking place in my country, Iran. It was during the Mahsa Amini protests that I wrote this thesis. It is my sincere wish that Iran and the Iranian people have a bright and peaceful future.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

As one of the most crucial yet challenging tasks in natural language processing, text generation has become increasingly important. Text generation, often formally referred to as natural language generation, is the task of producing readable and plausible human language text from input data, which can be a sequence of words [41]. There are various applications for text generation, such as machine translation[59], text summarization [26], and lyric generation for songs [69].

With the recent advances in deep learning, many researchers have developed deep learning models for the task of text generation. These models range from recurrent neural networks (RNNs) [11], Convolutional neural networks(CNNs) [22], Graph neural networks (GNNs) [40], and attention-based neural networks and transformers [19].

Deep learning models like RNNs and CNNs have difficulties in modeling the long-term dependencies in the text [31]. Transformer networks, on the other hand, are equipped with the self-attention mechanism. Self-attention allows every token to attend to all the tokens in the sequence and hence enables the network to learn long-term contexts [68]. This ability to learn long-term contexts, together with the capacity of learning from large volumes of text data has led to the huge success of Large Language Models (LLMs) based on transformers, such as GPT-2 [56], GPT-3 [8], and BERT[19].

Despite the fact that large language models are very proficient at memorizing and generating long sequences of text, they only have a limited ability to produce truly novel and creative texts such as poem lines. Alternatively, previously conducted research has shown that variational autoencoders (VAEs) have been successful in generating original poetry lines that adhere to the stylistic characteristics of the training corpus [69]. There are a number of factors that contribute to the origin and style of lines generated by VAEs. The first is the fact that VAEs can be trained on small highly curated corpora that reflect a certain style, and the second is the fact that VAEs with a recurrent neural network architecture has a relatively low memorization capacity, as opposed to transformer networks, which leads to them creating more creative texts.

1.1 Motivation and Problem Definition

The terms coherence and cohesion in linguistics are commonly defined as follows [71]:

- Cohesion: Cohesion is the semantic relation between one element and another in a text [27].
- Coherence: There are two conditions that must be met in order for a text to be coherent: it must be consistent with the context in which it is created, and it must be cohesive.

Achieving both coherence and cohesiveness in long natural language text generation is challenging for two main reasons. First, there is no well-defined formal specification of the coherence and cohesion of a text. The second issue is that there is no standard measurement model for either of the two properties [16].

Natural language generation methods that use neural networks rely heavily on large amounts of human-generated text for their training [13, 64, 23]. When judged individually, these models generate sentences that resemble those generated by humans, but they fail to capture the local and global dependencies among sentences, resulting in incoherent text [15].

There has been extensive research in the computational linguistics community on coherence and cohesion, especially prior to deep learning. Because there are no formal specifications for coherence and cohesion, many different concepts and methodologies have been developed, including Rhetorical Structure Theory [47] and other approaches to quantifying these concepts [5, 20, 29, 67]. However, there has been little prior study investigating coherence and cohesion with neural models in the context of long-form text generation [15].

As we discussed earlier, VAEs are a family of neural models capable of producing creative and new lines of text. However, due to their fewer trainable parameters, VAEs are limited to creating short texts at the sentence level as compared to LLMs, which can also produce longer texts. Therefore, VAEs can only generate independent lines of poetic texts, and cannot produce coherent and complete poems.

Despite the advances in the field of text generation, text coherence is still a challenging and less studied problem. Generating coherent text is specifically difficult for two main reasons. First, the coherence and cohesion of sentences in a text are not specified in any formal way. Second, there is no widely accepted model to measure the coherence [16].

In this thesis, we propose a new model of coherence scoring that allows the system to rank independent lines generated by a VAE and construct a coherent poem. However, our solution is not restricted to VAEs. The scoring model is based on BERT, fine-tuned as a coherence evaluator. We propose a novel training schedule for fine-tuning BERT, during which we show the system different types of lines as negative examples: lines sampled from the same vs. different poems. The results based on the annotated data show that annotators labeled poems constructed by this method to be more coherent than randomly sampled lines.

1.2 Contributions

The main contributions of this thesis are:

- We introduced a BERT-based model to evaluate poem coherency.
- We explored and analyzed different ways of labeling negative data to teach model coherency.
- We experimented with different ways of training the model and introduced a method that we found to be the best for teaching coherency to the model. Specifically, we propose a new method of training with negative sampling: Random from All songs (RfA) and Random from the Same song (RfS), which are used in an alternating pattern during training.
- We report the results of annotations of pairs of poems, where one is a poem composed of randomly chosen lines, while the other is a poem composed from the lines selected by our method. The annotation results showed that nearly 80% of annotators find poems generated by our method more coherent.

1.3 Chapter Outline

This thesis is organized into the following chapters:

- Chapter 1 outlines the motivation, problem statement, and main contributions.
- Chapter 2 discusses related core concepts that are important to this work.
- Chapter 3 includes important related and previous research on poetry generation and coherent text generation.
- Chapter 4 explains our approach to scoring the lyrics/poem coherence and generating coherent poems. This chapter also includes our experiments and results.
- Chapter 5 provides a conclusion and summary of this work, followed by future work.

Chapter 2

Background

In this chapter, we provide an overview of the topics that are important core concepts required for this thesis. The chapter outline is as follows. We start by introducing the field of Natural Language Processing (NLP) and the main questions and tasks in this field. Then we go through an overview of Neural Networks and the relevant members of this family to the topic of this thesis.

2.1 Natural Language Processing

Natural language Processing (NLP) is a sub-field of Artificial Intelligence (AI) that uses computational techniques to analyze and process large amounts of natural language data, with the final goal to enable computers to understand the content and language within written or spoken documents.

Because World Wide Web has a huge amount of data, at least 20 billion pages, NLP can be used as a valuable resource for finding important information from these pages, provided that important information can be found [18]. With the boost in Deep Learning in recent years, there has been great success in various NLP tasks and applications such as Information retrieval (IR), Information extraction (IE), Machine Translation, Question-answering (QA), Classification of text into categories, and Text generation [17].

2.1.1 Natural Language Generation

Natural Language Generation (NLG) is a sub-field of NLP that aims at building computer systems that can produce coherent and readable text [57]. NLG tasks take input and generate the intended text based on it. NLG tasks can be categorized into the following categories based on their different types of inputs [30]:

- **Text**: This paradigm processes unstructured text inputs to generate the text outputs. There can be various types of text inputs, including keywords, topics, prompt sentences, and sentiment labels [42, 54, 74].

TEAM	WIN	LOSS	PTS	FG_PCT	RB	AST	...
Pacers	4	6	99	42	40	17	...
Celtics	5	4	105	44	47	22	...

PLAYER	H/V	AST	RB	PTS	FG	CITY	...
Jeff Teague	H	4	3	20	4	Indiana	...
Miles Turner	H	1	8	17	6	Indiana	...
Isaiah Thomas	V	5	0	23	4	Boston	...
Kelly Olynyk	V	4	6	16	6	Boston	...
Amir Johnson	V	3	9	14	4	Boston	...
...	...	...	...	...	...	...	

PTS: points, FT_PCT: free throw percentage, RB: rebounds, AST: assists, H/V: home or visiting, FG: field goals, CITY: player team city.

The **Boston Celtics** defeated the host **Indiana Pacers 105-99** at Bankers Life Fieldhouse on Saturday. In a battle between two injury-riddled teams, the Celtics were able to prevail with a much needed road victory. The key was shooting and defense, as the **Celtics** outshot the **Pacers** from the field, from three-point range and from the free-throw line. Boston also held Indiana to **42 percent** from the field and **22 percent** from long distance. The Celtics also won the rebounding and assisting differentials, while tying the Pacers in turnovers. There were 10 ties and 10 lead changes, as this game went down to the final seconds. Boston (**5–4**) has had to deal with a gluttony of injuries, but they had the fortunate task of playing a team just as injured here. **Isaiah** Thomas led the team in scoring, totaling **23 points and five assists on 4–of–13** shooting. He got most of those points by going 14–of–15 from the free-throw line. **Kelly Olynyk** got a rare start and finished second on the team with his **16 points, six rebounds and four assists**.

Figure 2.1: An example of NLG from data input(table on left) generating the text summary (right). Retrieved from [55]

- **Data**: The paradigm generates new output text based on structured data input, retaining as much relevant information as possible. Graphs, expert systems, records databases, spreadsheets, and simulations of physical systems can all be used to represent non-linguistic data (such as knowledge-based or table-based data) [42, 10, 55]. Figure 2.1 shows an example from this category.
- **Multimedia**: This paradigm generates output text based on multimedia data input such as image, video, and voice. Examples of this category are image captioning [70] and song generation from music [69].

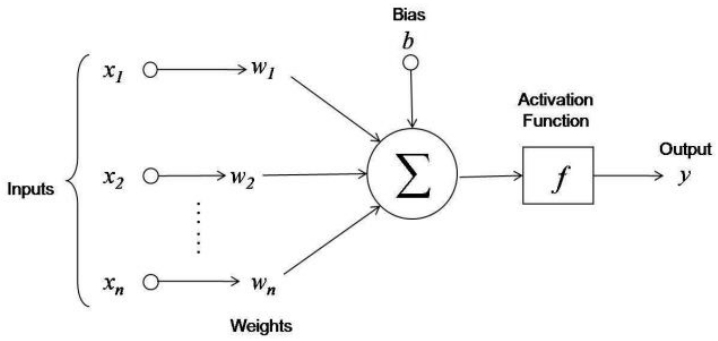


Figure 2.2: A single neuron in an artificial neural network. Source

2.2 Neural Networks

Neural Networks (NNs) are systems inspired by the neural connectivity of the human brain. They consist of groups of neurons connected to each other through synapses representing a weight. Each neuron takes the inputs from the neurons connected to it multiplied by their connection weight, and then sums these values up and adds a bias value to them. Then it passes this summed output through a non-linear activation function. There are a variety of choices for activation functions, including sigmoid, tanh, ReLU, and etc. Figure 2.2 shows the structure of a single neuron in a NN.

Neurons are aggregated in groups representing layers in the network. The neurons of each layer are connected to the ones in other layers. Feed-forward Neural Networks are the most common and basic type of NNs. In these networks, the information flows from the input through a set of hidden layers to the output layer in one direction. Figure 2.3 shows an example of a NN that all of the neurons of each layer are connected to the previous and next layer neurons. This network is called fully connected.

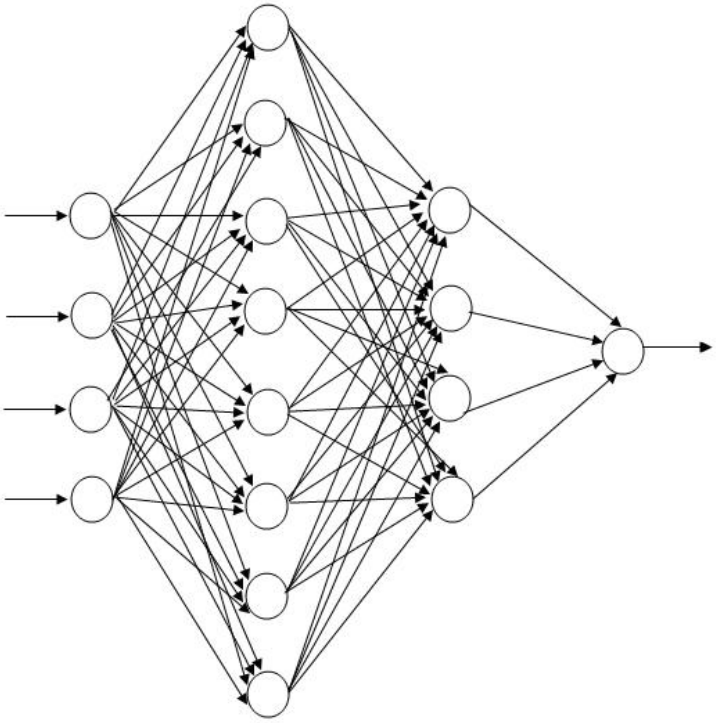


Figure 2.3: A fully connected feed-forward Neural Network consisting of two hidden layers. Source

2.3 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a class of artificial NNs that have connections that create cycles from the outputs affecting the inputs to the neurons. These cycles, alongside the state memory, allow RNNs to compute sequential dependencies in sequences of arbitrary length. It is also important to note that RNNs can cover the full sequence of input data as they unfold many times, so the input can be as long as desired.

Figure 2.4 left shows the structure of an RNN with one hidden layer. The RNN unfolds in time to process all timestamps in the input sequence which results in the misleading appearance of layers. Although RNNs can consist of several hidden layers, these layers should not be mistaken by the unfolded network where the output gets propagated backward to influence the next time-stamp's input.

The ability to process sequence data makes RNNs a good fit for NLP tasks where the input is usually a sequence in form of words and sentences. RNNs and variants of them

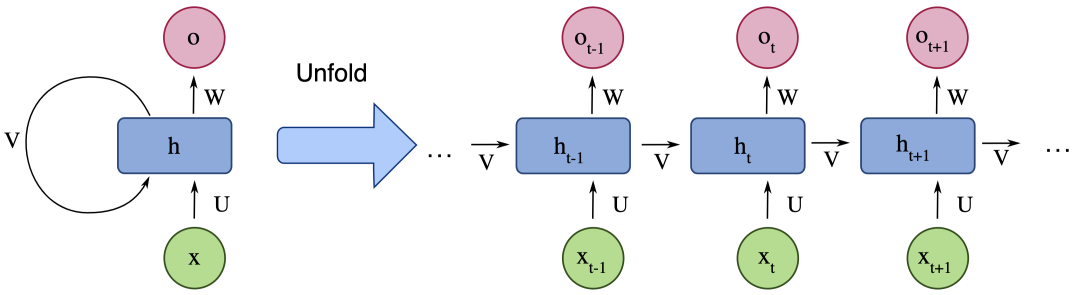


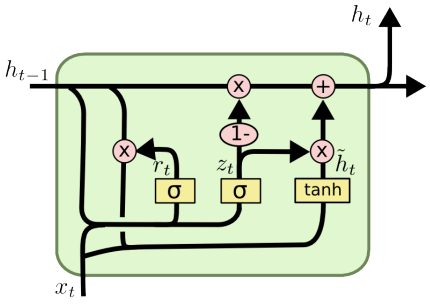
Figure 2.4: A fully connected Recurrent Neural Network. x denotes the input, o denotes the output, h is the hidden state, v and w and u are the weights for the feedback, input, and output connections respectively. Source

have demonstrated considerable performance in different speech recognition [60, 25, 24, 61] and language modeling tasks [49, 63, 45, 65].

Although training feed-forward NNs is straightforward via the gradient descent, RNNs face difficulties while training due to the feedback loop that causes problems in learning long-term dependencies. In a dynamic learning process, the gradients of a hidden state at later time stamps are sensitively dependent upon those at the beginning, so their values can exponentially grow and cause an unstable learning process (which is called the exploding gradient problem). The gradients can also converge to zero if the gradients in earlier steps are smaller than one (this is called the vanishing gradient problem) [6, 48].

2.3.1 LSTMs

Long Short-Term Memory (LSTM) [28] is a type of RNN equipped with additional gating mechanisms to overcome the challenges in training RNNs for capturing long-term dependencies. LSTMs have input, output, and forget gates in addition to the state memory. Figure 2.5 shows the structure of an LSTM cell alongside the gating computations. The output of the previous timestamp loops back to the input of the next timestamp.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 2.5: An LSTM cell. x is the input and h is the hidden state. W denotes the weights. Source

A growing number of industry and academic applications rely on LSTMs to perform time series classifications and predictions. There are a variety of tasks that are involved

in this process, including speech recognition, sentence embedding, and correlation analysis [38].

2.4 Autoencoders

Autoencoders [4] are a type of neural network designed for dimensionality reduction via learning compact coded representations from input data. The autoencoder consists of an encoder part, a code, and a decoder part. The encoder part compresses the input into the code (also called the hidden representation or latent space) through a number of hidden layers. The decoder part takes the code and reconstructs the input from it. Figure 2.6 shows the structure of an autoencoder network.

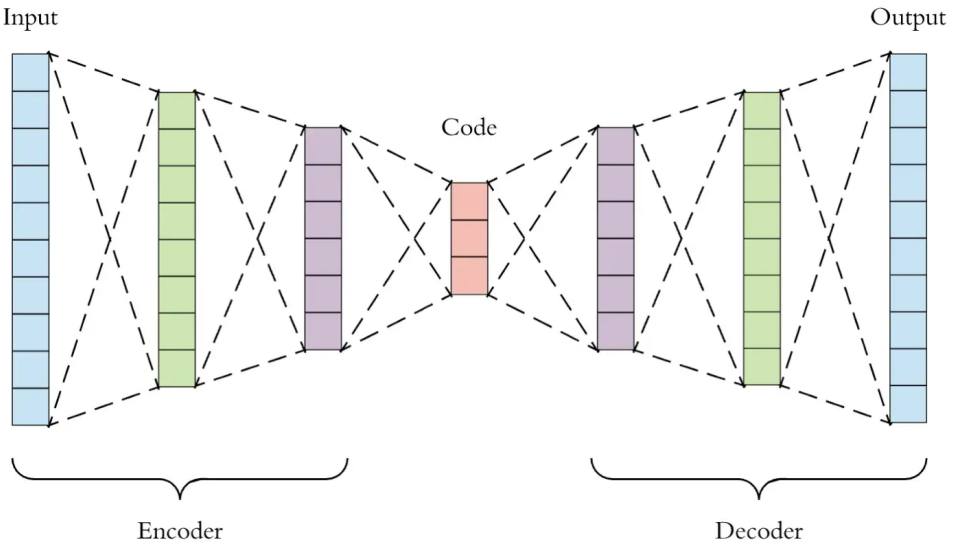


Figure 2.6: An Autoencoder consists of three parts: encoder, code, and decoder. Source

The hidden layers in autoencoders can be simple fully-connected layers, Convolutional layers, or RNN layers. Autoencoders are relatively simple to train since they do not require labeled data for training and they use the inputs as the label. This important feature makes them popular to learn the hidden features in data. The code extracted by the autoencoder contains the most important information present in the input in a compressed version which can be used to store small-sized representations of the data. Moreover, the autoencoder can be trained with classification at the same time to use the code for classification.

2.4.1 Variational Autoencoders

When trained properly, the autoencoder can be used as a generator model by using the decoder part with randomly sampled code input. Like a standard autoencoder, a variational autoencoder [33] also consists of an encoder and decoder and trains to minimize the reconstruction loss between input and decoded data. The key idea is that rather than encoding an input as a single point, the variational autoencoder encodes it as a distribution over the

latent space. By sampling from the learned prior distribution, the decoder generates novel data samples during the inference phase. Using Kullback-Leibler (KL) divergence [35], the posterior distribution is pushed closer to the prior distribution.

In recent years, variational autoencoders have gained popularity in the scientific community due to their strong theoretical probabilistic foundation and valuable insights into latent representations [62, 2].

2.5 Transformers

Transformer [68] is a NN model that adopts the self-attention mechanism primarily used in the fields of NLP and Computer Vision. The self-attention mechanism computes a representation of a sequence by relating different positions of the sequence. In addition to learning task-independent sentence representations [44], self-attention has been successfully employed to improve reading comprehension [51], abstractive summarization [52], and textual entailment [12].

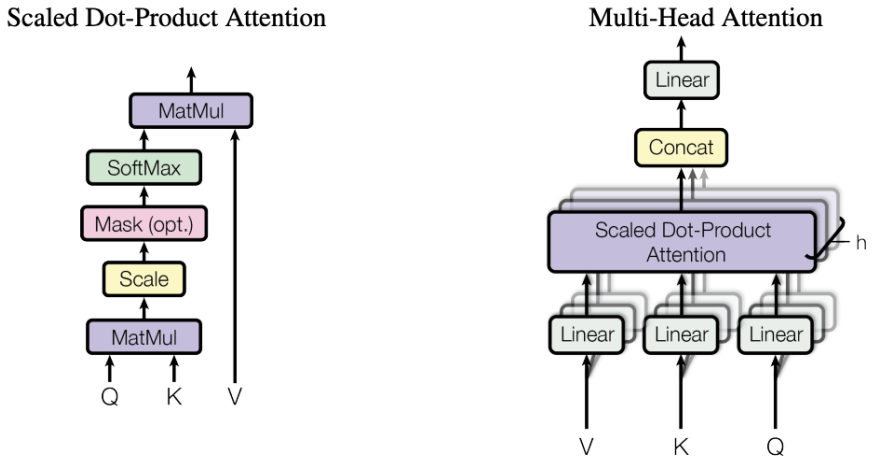


Figure 2.7: Scaled Dot-Product Attention(left). Multi-Head Attention(right). Source: [68]

An attention function is essentially a mapping between a query and a set of key-value pairs to an output. All of the elements are vectors, including the query, the key, the value, and the output. Using a compatibility function of the query with the corresponding key, the weights assigned to each value are calculated and the output is computed as a weighted sum. Transformers use scaled dot-product attention in several layers running in parallel, forming multi-head attention depicted in Figure 2.7.

Figure 2.8 shows the Transformer architecture. The transformer has an encoder-decoder structure similar to several neural sequence models [14, 3, 64]. The transformer encoder consists of N identical layers, each containing two sub-components: The multi-head attention, followed by a simple, position-wise fully connected feed-forward network. The structure of the decoder is similar with an additional multi-head attention layer processing the encoder output. Moreover, there are residual connections present in both the encoder and decoder parts of the transformer.

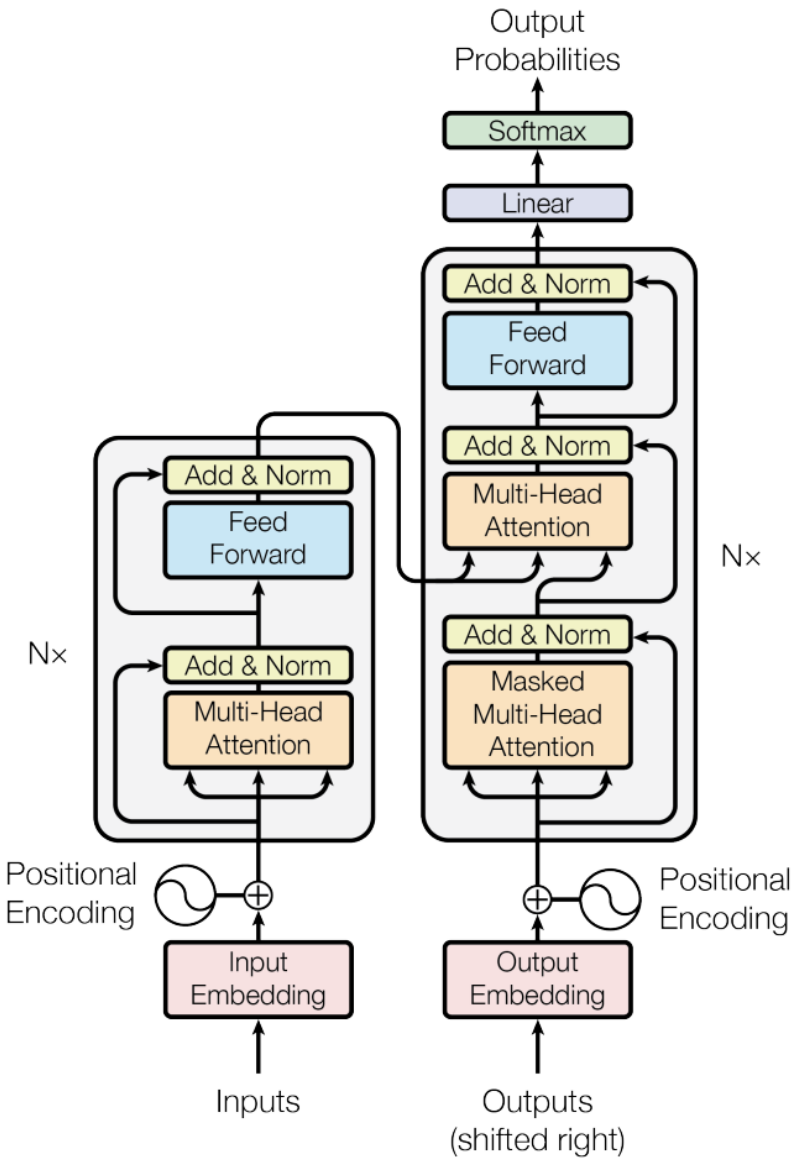


Figure 2.8: A Transformer Neural Network. Source: [68]

In recent years, NLP has embraced transformers as the preferred model to process natural language. Pre-training on a large corpus of text is common, followed by fine-tuning on task-specific data [19].

2.5.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) [19] is a transformer-based machine learning technique for NLP pre-training. BERT has an almost identical structure to the Transformer architecture. In all layers of BERT, left and right contexts are jointly conditioned to pre-train deep bidirectional representations from the unlabeled text. Therefore, it is possible to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, by adding just one additional output

layer to the pre-trained BERT model.

Chapter 3

Related Work

In this chapter, we outline the research works that are closely related to the contributions of this thesis. We first introduce the poem generation model that was used as our base generator. Then, we provide a comprehensive review of the NLG coherence literature.

3.1 Lyrics Generation

This work uses LyricJam [69] as the backbone model for generating poem lines. Based on the audio of live music played by a musician, LyricJam generates lyrics in real-time using two approaches: GAN-CVAE, which adversarially predicts lyric representation using music representation, and CVAE-spec, which transfers the VAE's spectrogram latent space to the text CVAE's lyric latent space.

Using VAEs, LyricJam creates novel and creative poem lines. Its goal is to generate individual lyric lines as sources of inspiration for a musician, instead of complete poems or song lyrics.

3.2 Coherence in NLG

One of the most important and fundamental research fields in the generation of text output is text coherence. The difference between a coherent document and a random collection of sentences lies in the fact that its components have a logical relationship. Many theories have been proposed in recent years to evaluate coherence in texts and to create systems that produce texts that are very close to the ones written by humans [1].

3.2.1 Coherent Text Generation

In the field of text generation, defining an ideal loss for training models remains a work in progress. Many works based on RNNs use cross-entropy loss for training [3]. Additional terms may be used to cover specific topics or supervise the model on specific tasks [73, 32].

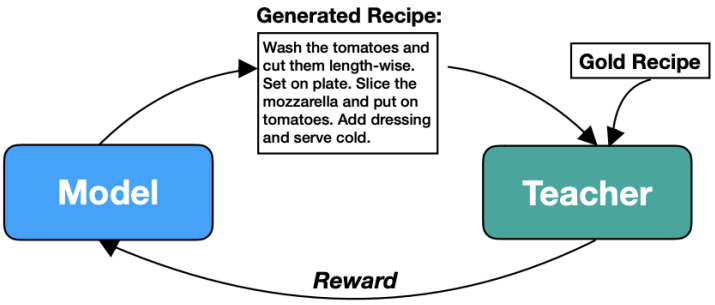


Figure 3.1: Rewarding the NLG model by a teacher model to imitate the discourse structure of reference sequence. Image taken from [7]

Research has shown that training with cross-entropy does not always lead to achieving good scores on common evaluation metrics such as ROUGE [43] or BELU [50]. Therefore, reinforcement learning-based [72] training generation models are also being investigated that try to directly optimize the target evaluation measures [58, 53].

It is important to note that the majority of automatic measures rely on local n-gram patterns, offering a limited and myopic perspective on overall text quality. Thus, even though models trained to directly optimize these measures can yield better results on these measures, they may not improve the overall quality of the text in terms of coherence or discourse structure [7].

To overcome the problem with generated text coherence, Bosselut et.al. [7] propose a method to learn neural rewards and use them in a reinforcement-learning algorithm with a specific focus on coherent text generation. Their reward learning scheme approximates the desired discourse structure in documents by capturing a cross-sentence ordering structure. Using self-critical reinforcement learning, the learned teacher calculates rewards for the underlying text generator. Figure 3.1 shows their proposed method.

Cho et.al. [15] proposed a language model with two discriminator networks. They provide feedback signals to the generator at two levels: word level for sentence cohesion and sentence level for paragraph coherence. They train their model with a negative-critical sequence training policy gradient. The coherence discriminator is designed to distinguish a coherent text chunk of sentences from an artificially built incoherent chunk.

Kiddon et.al [32] proposed a model for globally coherent text generation called the neural checklist model. It uses RNNs to model global coherence by storing an agenda of text strings that must appear in the output at some point. Using a language model and two attention models that encourage agenda references, the model generates output by dynamically interpolating between them.

3.2.2 Coherence Evaluation

Due to the open-ended nature of many NLG tasks, evaluating their output is challenging. Given an NLG task, the system can generate multiple plausible outputs for the same input. For example, different lyrics can be generated for the same music played. As a consequence, human evaluation remains the gold standard for the majority of NLG tasks. In spite of

this, human evaluation is expensive, so researchers often rely on automatic metrics as a means of evaluating daily progress and optimizing systems on an ongoing basis [9].

The evaluation methods for NLG can be categorized into the following categories:

- **Human Evaluation** Humans are the best judges of the quality of text generators. A Turing test [66] is used to distinguish machine-derived texts from human-derived texts when naive or expert subjects rate or compare texts generated by different NLG systems.
- **Untrained Automatic Evaluation** Automatic metrics are among the most commonly used categories in research. A machine-generated text is compared to a human-generated reference text based on the same input data and using metrics based simply on string overlap, content overlap, string distance, or lexical diversity, such as n-gram match and distributional similarity, without the need for machine learning [9].
- **Machine-learned Automatic Evaluation** It is often the case that these metrics rely on machine-learned models to measure how closely two machines generate texts or how closely two machines generate texts compared to human-generated texts [9].

Machine-learned Coherence Evaluation Tasks

The NLP community has proposed different models to measure the coherence in text documents and also a set of tasks to evaluate the performance of these models [36]. These tasks can be outlined as follows.

- **The Insertion Test** is a test in which the model should predict the correct location of a sentence removed from a text document. This is usually done by scoring each sentence position and finding the one with the highest score. The problem with the insertion test is that it is typically considered a difficult task and the models can not go beyond 10-20 percent of accuracy. Moreover, it might be a difficult task even for humans since there might be several correct positions for a given sentence.
- **The Sentence Reordering Test** [46]. In this task, the model should reorder a shuffled text into the original form. Since it's too expensive to score all combinations of sentences to find the best sentence ordering, this task is limited to generative models [36].
- **The Shuffle Test** [5] is one of the most common tests to evaluate the model text coherence. It is a binary classification task, in which the model should learn to correctly label shuffled and un-shuffled text documents. The shuffled text is obtained by randomly changing the order of the sentences in the original document. One of the most common datasets used in this task is the dataset of articles from the Wall Street Journal [21].
- **The k-block Shuffle Test**. With the appearance of large transformers, the shuffle test can be considered a solved task. Laban et.al. [36] introduced a new test, called the k-block shuffle test where instead of reordering the sentences in the text, they reorder the blocks of text. They show that this test is more difficult for the models, dropping their accuracy from 94% to 77%.

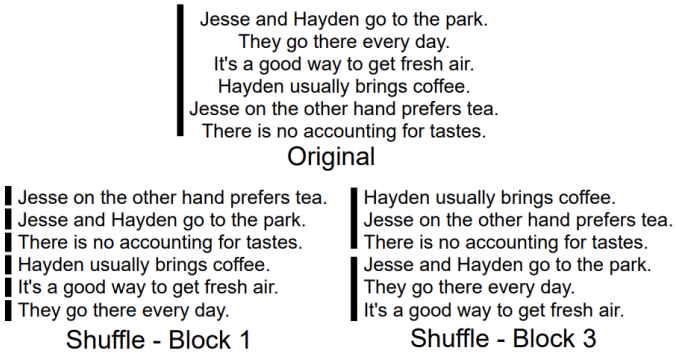


Figure 3.2: Shuffled text in shuffle test (left) and k-block shuffle test (right). Image taken from [36]

3.2.3 Coherence and Creativity

NLG models can learn to copy samples from a training dataset and produce samples that a human judge will deem to be of high quality, but may not be able to generate diverse samples (e.g., samples that are very different from the training samples), as has been observed for social chatbots [39, 75]. When a language model is optimized solely for perplexity, the responses may be coherent but bland. This behavior usually occurs when generic large pre-trained language models are used for downstream tasks without fine-tuning on the datasets that are task-related [9].

Chapter 4

Method

In this chapter, we introduce our method and results. First, we introduce the dataset we used in this thesis. Then we describe our model and its goal. We continue by describing our training pipeline and model evaluation. Finally, we present the evaluation results.

4.1 Dataset

This thesis uses a dataset that consists of approximately 57,000 songs from different genres that were produced prior to 2016.

The number of lines in each song in this dataset forms a right-skewed normal distribution with a mean of 35 lines. The histogram showing the number of lines in these songs is depicted in Figure 4.1

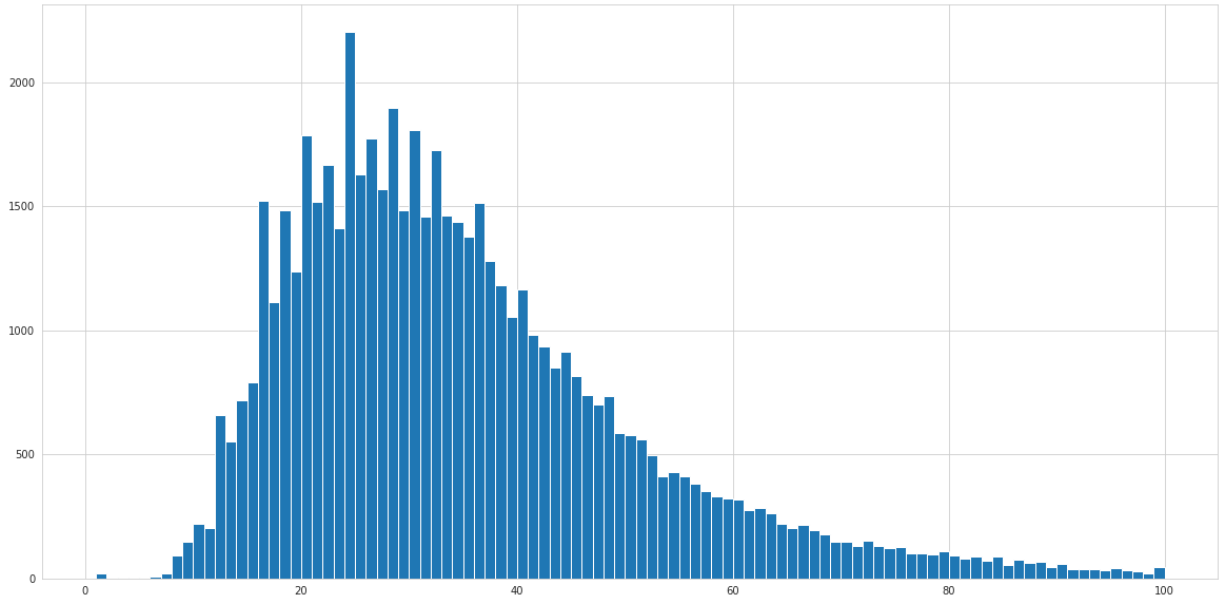


Figure 4.1: Hist diagram of the number of lines

The dataset contains two million tokens in total. The number of tokens of each line in this dataset forms a right-skewed normal distribution with a mean of 7 tokens. A histogram of the number of tokens in the lines is depicted in Figure 4.2

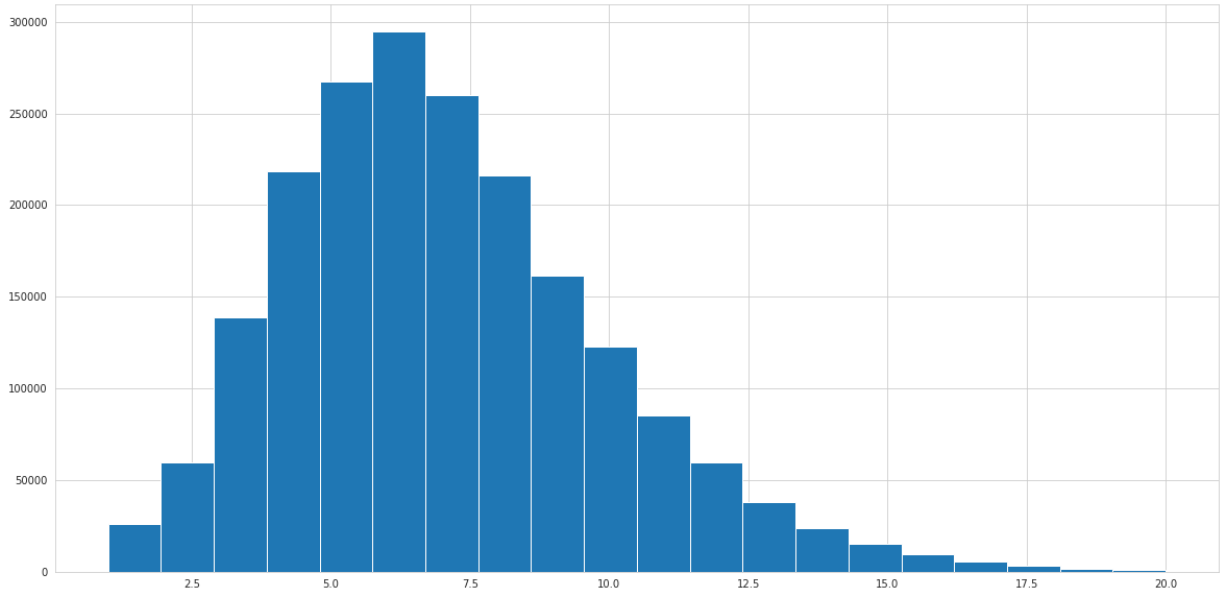


Figure 4.2: Hist diagram of the number of tokens in each line

4.2 Goal of the model

We first aim to develop a model to measure the coherence of a poem by assigning coherence scores to the inputs. The input to this model would be a number of consecutive lines already generated, alongside the new lines generated by the generator model. The ultimate goal would be to train the measuring model in a way that gives higher scores to the best-fitting lines and low scores to the lines that do not match the previous lines. Figure 4.3 shows a schema of the performance of this model on an example. In this example, we expect the model to give a low score to the line "The dog is here", indicating that the model finds this line to be incoherent in comparison to the previous lines. On the other hand, "Oh, I believe in yesterday" would get a high score, which indicates that it is the best of these lines based on the model.

In order to build this measuring model, it is essential to first identify a criterion for coherence. The main question that arises here is how we should proceed to teach the model to properly score the input lines based on the overall poem coherency. In the next sections, we will introduce our method to evaluate and measure coherence.

4.3 Coherency Criterion

There is no formal specification of the coherence and hence, no universally accepted measurement model for quantifying the coherence of a text. However, it can be argued that

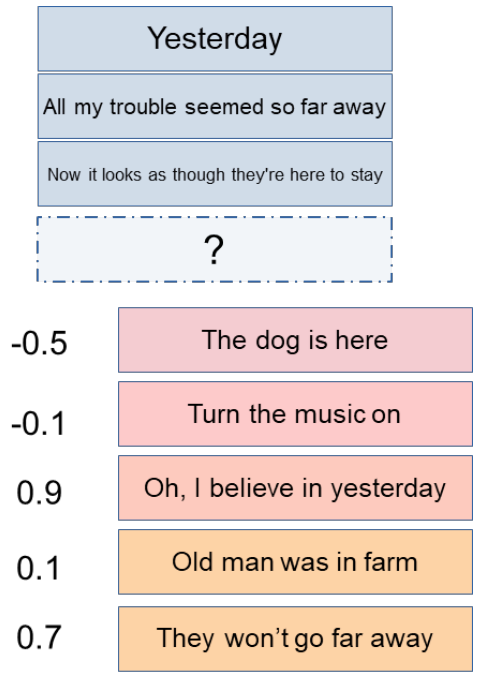


Figure 4.3: example of model scoring the last lines

since coherence appears in human writing and poetry, it would be sufficient to generate incoherent samples and supervise the model to score the coherent and incoherent samples accordingly. In the task of measuring coherence in poems, we train our model to correctly classify input poems into coherent and incoherent classes. After training the model on this binary classification task, we expect the model to be able to properly score the coherence of the last line according to the previous lines.

We first need to build the dataset with coherent and incoherent labels. Coherent labels are already available through human-written poetry. In order to create the incoherent class samples, we perturb the coherent samples by changing the last lines of these samples. In this way, we will have samples that only their last lines are incoherent, which is exactly what we want our model to focus on when scoring the inputs.

There are several ways to change the last line of the poems to create the perturbed samples. However, it is important to note that not all changes are proper for our task, which is creating incoherent samples. In order to properly teach the model coherency, the only difference between the samples with positive and negative labels should be the coherency and not any other factor. Consider the scenario in which random nonsense characters are substituted for the last line. In this case, the model will probably be able to distinguish between the labels more easily by learning whether the last line words are meaningful rather than learning if they are coherent.

We studied two ways of generating incoherent labels in this thesis which will be described below.

4.3.1 First Method: Random from All songs (RfA)

In the first method, we replaced the last line of a real song with another line randomly selected from all songs in the dataset. For example, consider a hypothetical dataset consisting of three songs as depicted in Figure 4.4.

Song 1	Song 2	Song 3
You are the dancing queen	Yesterday	Like the legend of the Phoenix
Young and sweet	All my trouble seemed so far away	All ends with beginnings
Only seventeen	Now it looks as though they're here to stay	What keeps the planet spinning
Dancing queen	Oh, I believe in yesterday	The force from the beginning
Feel the beat from the tambourine	Suddenly	We've come too far
You can dance	I'm not half the man I used to be	To give up who we are
You can jive	There's a shadow hanging over me	So let's raise the bar

Figure 4.4: Hypothetical dataset

Then two possible "coherent" and "incoherent" labels are in Figure 4.5

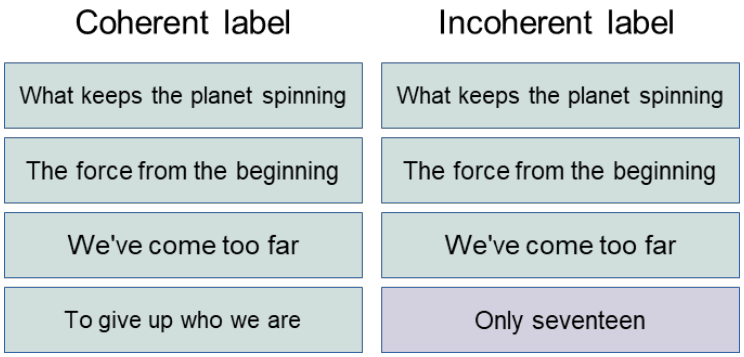


Figure 4.5: Hypothetical RfA labels

Despite its advantages, this method has been subject to some criticism. Although positive and negative labels can be distinguished based on their coherence, other factors differentiate them too. It is expected that the last line in the false labels will differ from the previous lines in phonetics, semantics (meaning), and grammar (structure) since we are selecting the last line randomly from all the songs. The model can use these features to differentiate between the labels rather than coherence. Due to this, our model may not learn and score coherence but instead learn and score other features.

4.3.2 Second Method: Random from the Same song (RfS)

In the second method, to generate incoherent last lines, we replaced the last line of an actual song with another line randomly chosen from that exact song. We managed to avoid picking a line already present in the window size. For example, consider our dataset of songs to be depicted in Figure 4.4, then two possible "coherent" and "incoherent" labels are in Figure 4.6.

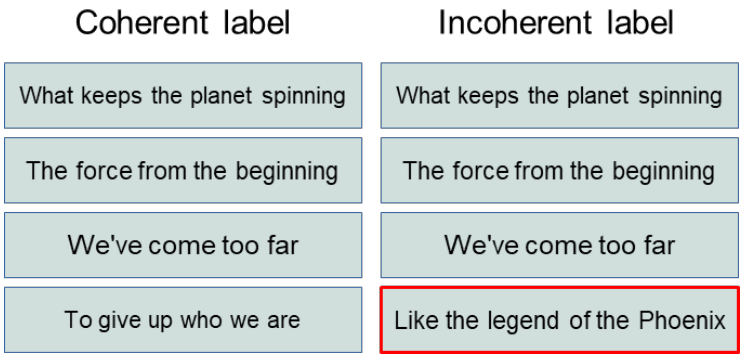


Figure 4.6: Hypothetical RfS labels

The majority of the songs have duplicate lines, so by picking a line from another line of the song, the same line could be selected. It is important that the randomizer takes care of this and only selects the same line once. As a result of this method, it is more difficult to differentiate between labels without considering coherence. Consequently, a

model that is successful in distinguishing between these samples in this method is more likely to have a better understanding of coherence. As we will see in the future, this method has the disadvantage of being difficult to train. Compared to the first method, it is a more challenging task. Human annotations indicate that it is even difficult for a human to distinguish this type of labeling.

4.4 Training the Model

This thesis uses Base BERT as a pre-trained language inference model. We will fine-tune BERT to categorize whether a given input is coherent.

It is essential first to define the meaning of "window" and "window size" in this thesis. "Window size" refers to the number of lines provided to the BERT model in each input. "Window" refers to a continuous sequence of lines taken from a song. In each case, the BERT will be given a fixed number of lines to determine whether the last line belongs to the others. For example, in Figure 4.7, both "coherent" and "incoherent" labels with a window size of four can be seen.

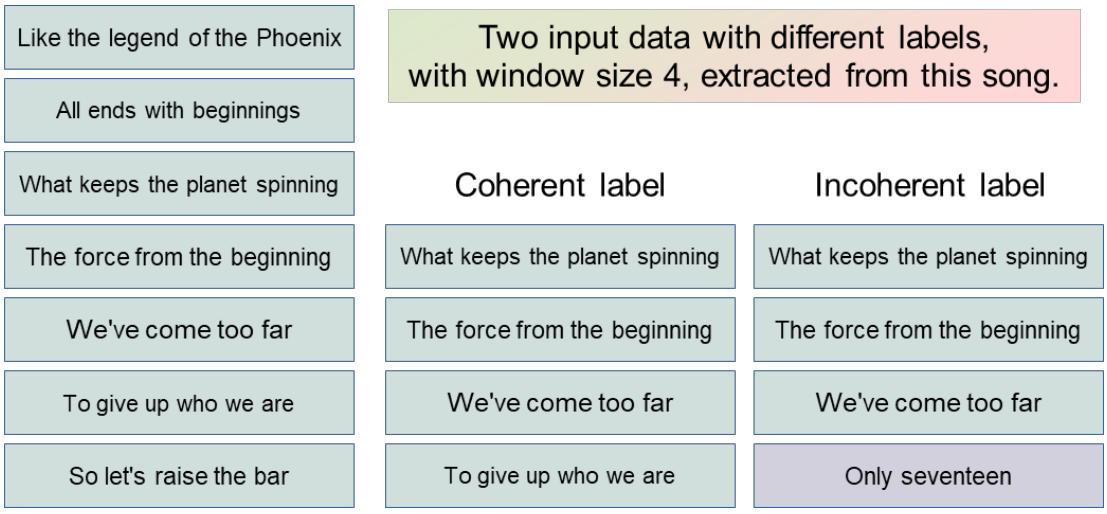


Figure 4.7: Hypothetical four-line window with RfA label

The lines in the window were first concatenated with BERT's [SEP] tokens. A [SEP] token refers to a separation token. After passing it to the BERT, we categorized the input label with the binary one-hot output after passing its output from a fully-connected layer. The process is depicted in Figure 4.8.

4.5 Labeling with the First Method (RfA)

Using the first method RfA, we labeled the incoherent data using the last line of the window selected randomly from the entire dataset. There will be only one window extracted for each song in the dataset. The window is chosen randomly from the song. There is a fifty-fifty chance that the window will be labeled as either "coherent" or "incoherent". When

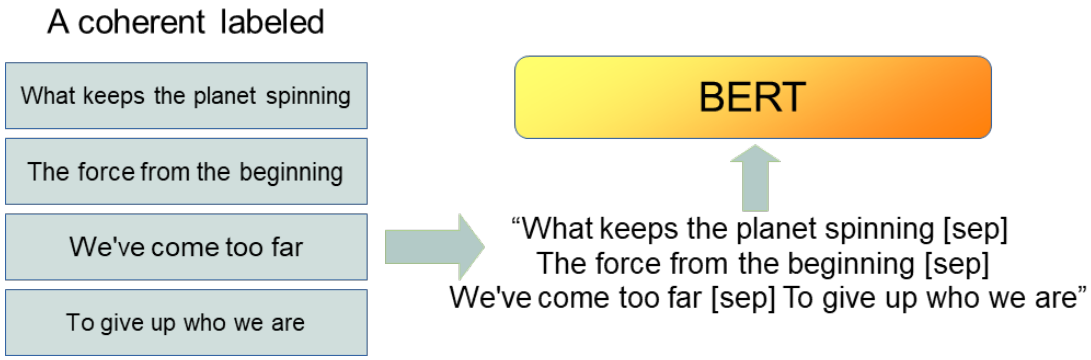


Figure 4.8: example of BERT input

the window is deemed “incoherent”, the last line will be removed and replaced randomly. After training with different window sizes, we achieved the results shown in Table 4.1.

Window size	Accuracy
3	76
4	77
5	79
6	80

Table 4.1: Accuracy of Models with Different Window Sizes.

By increasing the window size, it can be seen that accuracy increases as well. It is because, for the model to detect the labels, it must determine whether the last line is the ground truth line in the window. It is easier to judge lines with a larger window because they contain more lines. An example would be that in a window with three lines, the last line should be judged only based on two lines, which would be more complicated than judging five lines in a window with six lines.

An important question arises here. If we were to randomize not only the last line but the two last lines or even more than that, what would be the accuracy of the results? We experiment with this on a four-line window by constructing negative data samples where all four lines are replaced with random lines. As a result, the accuracy grew to 87 percent, 10 percent higher compared to replacing only the last line with random. This process is visualized in Figure 4.9. Green boxes with a “T” indicate actual (ground truth) lines, while red boxes indicate randomly sampled lines. The top row in each image represents a positive sample, while the bottom row represents a negative sample.

Figure 4.9: Window size four incoherent labels: All lines randomized vs. the last line randomized.

Even though the accuracy of randomizing all lines is much higher, we cannot use this labeling method due to our problem’s nature. The goal is to train the model to select the

best line that fits the previously generated lines. However, this finding opens up a new avenue for improving accuracy. The topic is discussed in the following sections.

4.5.1 Cascade training

In this method, we teach the model to learn step-by-step rather than simply judging the last line right away. Step-by-step here means the number of random lines ("F"s in 4.9) decreases in every two epochs. We first teach the model to distinguish if a window is entirely true or if all of its lines are randomized for two epochs. Then we will decrease the number of randomized lines by one for the following two epochs until there remains only one randomized line, the last line. The process for a four-line window is visualized in Figure 4.10.

Figure 4.10: Cascade training phases in a four-line window

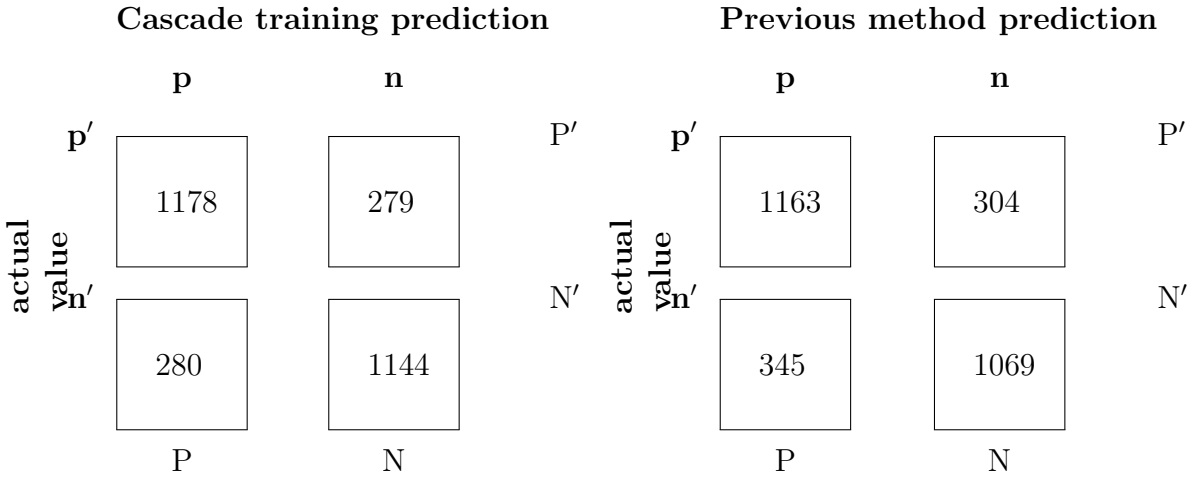
Three phases are depicted in Figure 4.10. Each phase will be trained for two epochs. Note that phases will transition from four "F"s to two "F"s because there is no difference between an entirely random window and a window with only the first line being a "T". The result of cascade training is a 3% increase in accuracy for every window size. The accuracy in each phase is shown in Table 4.2.

Window	1	2	3	4	5	Cascade	Not Cascade
2	71						
3	80	75				75	76
4	87	86	80			80	77
5	91	90	88	81		81	79
6	94	93	90	89	83	83	80

Table 4.2: Cascade training results in every phase vs. previous results.

Further analysis shows that the Cascade training method decreases both true negatives and false positives. Both methods' confusion matrices are shown in Table 4.3.

Table 4.3: Cascade training confusion matrix vs. previous method confusion matrix



4.5.2 Starter Model

A four-line window model cannot select the next line without three previous lines. In the same way, a five-line window model cannot function without four initial lines. Therefore we need another model, a starter model, to choose $WindowSize - 1$ initial lines. Unlike our main model, instead of judging the coherency of the last line by looking at the lines before, the starter model judges whether all given lines are a coherent start to a song. A starter model is built similarly to the first phase of Table 4.2. The only difference is that the correct labels are only selected from the beginning of the songs rather than from the middle.

The training results of the three-line window starter model and the four-line window starter model can be found in Table 4.4. Larger window starter models are not trained

since they are not feasible, as explained in the next section. In a case where, for example, we require seven initial lines for an eight-line window model, we could generate four initial lines with the starter model and then use a five-line model to expand these initial lines to eight.

Window size	Accuracy
3	89
4	91

Table 4.4: Accuracy of starter models with different window sizes.

4.5.3 Using the Cascade Trained Model

The coherency scoring model aims to select the best combination of coherent lines from a generative model. In this thesis, we used LyricJam as the generative model [69]. LyricJam produces poem lines for each timestamp of a given melody input. We asked LyricJam to generate ten lines for each timestamp then we used the coherency scoring model to select a sequence of lines. As input, we used 10-second clips of instrumental music[1]. Table 4.5 shows examples of lines generated by LyricJam in three consecutive timesteps. In each timestep, a 10-second audio clip from an instrumental music composition was fed to LyricJam, which returned ten lyric lines, generated conditioned on this audio clip.

Timestep 1	Timestep 2	Timestep 3
all i've still found	all of love is running out of me	all i see
i'm just caught a open mirror	i'll never find the light	i'm a lover in the moon
but if you felt still home	i'll drown away	now when you're lost
the clouds has their way	when everything is more	people i've free
this lights without the sun	all of time of time	keeps it go away
lost the light	i can not the answers	what you drown
i don't let you get away	all i drown	the walls will let the end
stay ones we'll win lovers	all in return of you	assimilation you drown open
i'm never felt clear here	all is running out	when the world is passed
oh though it's still alive	now this is all you need you	and those thoughts in two

Table 4.5: Lines generated by LyricJam for three consecutive 10-second audio clips.

There are a variety of methods that we can use for this model to select lines. These methods are discussed further in this thesis. However, in this section, we only discuss the greedy method. In the greedy method, the model chooses the line with the highest predicted coherence probability for poem continuation. In Figure 4.11 greedy method is depicted for a four-line window size model. The model chooses the following line in a four-line window by looking at the three previously selected lines. In Figure 4.11, the model first selected "Line 02", which was predicted to be the most coherent line based on

[1]Ghosts I-IV. Nine Inch Nails. Produced by: Atticus Ross, Alan Moulder, Trent Reznor. The Null Corporation. 2008. Released under Creative Commons (BY-NC-SA) license

the three previous lines. Then the model moved its window and continued to select the following lines greedily. Nevertheless, to begin with, the model needs "initial lines", which are shown in the Figure.

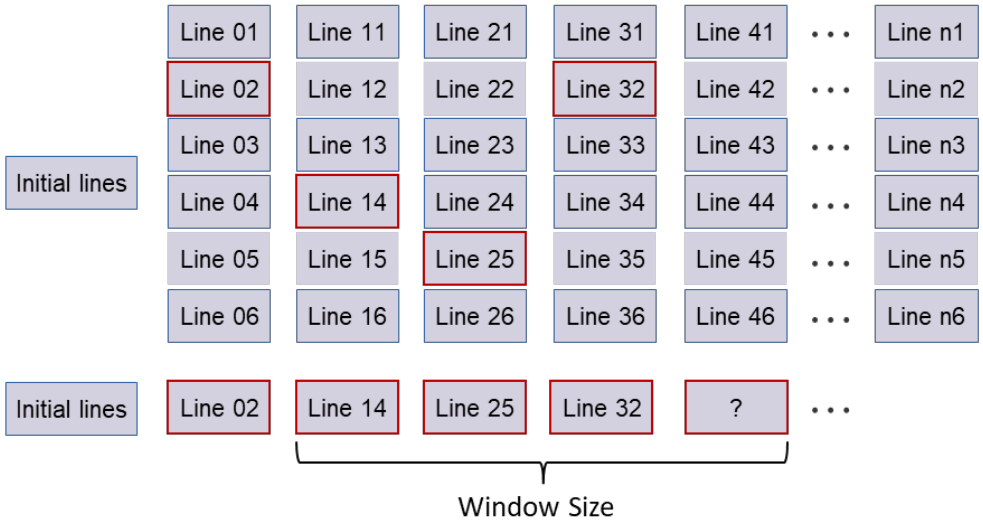


Figure 4.11: Four-line window model selecting lines with greedy method from each timestep

The initial lines are selected using the starter model. The starter model evaluates all possible combinations of the lines in the first n timesteps to choose the best. For example, a three-line starter model (n=3) evaluates a 10^3 combination of lines and chooses the most coherent sequence. Note that all combinations of starting lines grow exponentially with n; therefore, using a five or six-line starter model is not feasible. Below are three examples of ten-timestep-generated poems. Each example has two columns, one of the columns consists of randomly selected lines generated by LyricJam, while the other contains line sequences selected using a four-line model. The readers are invited to evaluate which poem is more coherent in each pair. The answers can be found immediately following the examples.

Example 1

when we'll only get away
a this place we will control
the ghosts of the moon
oh now i will lose you know
what you know
and this is still away
we will leave in the end
lost the light
the oceans keep me
all the light is not so cold

lost when you feel away
lost when it's miles away
when you get away
you give from another light
and this is still away
we will leave in the end
lost the light
all the things are not there
and we will end
and the glow of the moon

Example 2

i'll get all our place man
i'll get this to end
and when it's still here
it's the time to me
it's all our place
this's the sun
keeps us to fall
it's my dream
to feel your mind
keeps us by another end

the ghosts
the movie fools to come
all those it will have to fall in
stay the fools
the ghosts eaves away
we are fools with the fools
in the sun
i'm trying to fall
i got a ship of fools
i know my soul's falling

Example 3

now i keep inside
i leave in the ocean the soul
broken away inside
i don't let you get away
doesn't want another know you feel
if you go away
i was the moon
they have to let you go away
we come walls of fools
all i've undergone

the time is gone
the sun has lost the end
now the world is gone
broken away inside
here on the moon
now the world is just the only
love that's time
behind the world is gone away
falling in the moon
time away far away

Answers: Poems created using the coherency evaluator model are Right, Left, and Right. Accordingly, random poems are on the other side.

4.5.4 Problems with RfA labeling

Although poems generated using the RfA labeling method are more coherent than random selections, when analyzed, they have a problem. The RfA method understands the concept of coherency with a significant emphasis on word repetition. That is why we can see many repeated words in its generated songs. It tends to pick a line with the most number of common words with the previous lines. The following example illustrates how much repetition there is in an outcome of the RfA method.

lost when you feel away
lost when it's miles away
when you get away

you give from another light
and this is still away
we will leave in the end
lost the light
all the things are not there
and we will end
and the glow of the moon

Analyzing some actual song inputs shows its bias toward word repetition. Table 4.6 shows RfA gave the score -1.4 to an actual song "No time to die" by the singer Billie Eilish. This is a low score, meaning the model does not find it coherent. Nevertheless, if we replace the last line with only some words from the previous lines, the model will give it a score of +4.4. This is an excellent score for this low-quality selection that was made and is only due to the repeated words.

	Lyric lines	Score
Real	I should have known [SEP] I'd leave alone [SEP] Just goes to show [SEP] That the blood you bleed is just the blood you owe	-1.4
Fake	I should have known [SEP] I'd leave alone [SEP] Just goes to show [SEP] know leave alone show	+4.4

Table 4.6: RfA cascade training scoring examples.

This behavior may be due to the labeling. When we randomly choose lines from all songs for negative ("incoherent") samples, the line that will be chosen does not have common words or semantic fit with the previous lines. So the model can distinguish "incoherent" labels by just watching for common words. Because if there are common words, it is most probably a "coherent" label.

Even though having common words can be a sign of coherency, it is only sometimes the case. For example, if we take a poem and shuffle its lines, it would not be considered coherent, although the number of common words between lines has not been changed.

We tested the model to see if it can identify incoherent poems that are generated by replacing the last line randomly from the same song (RfS labeling). It is similar to shuffling the lines of a poem to make it incoherent. The experiment shows that by RfA training, the model performs almost as poorly as a random selection on an RfS test dataset. As shown in Table 4.7, the accuracy drops from 80% to 55%.

Window 4 model	Accuracy on RfA	Accuracy on RfS
RfA training	80	55

Table 4.7: four-line window RfS labeling method accuracy on RfS and RfA test datasets

We can conclude that RfS training is not a suitable method of teaching model coherency because it can not distinguish RfS types of incoherency. This brings us to a new idea that will be discussed next — the second method for labeling.

4.6 Labeling with the second method (RfS)

The second proposed method for labeling is to replace the last line with a randomly selected line from the same song. This method sounds more intuitive to teach coherency to the model than RfA, but it is also more challenging for the model to learn.

Training the model using the cascade method discussed previously shows a boost in RfS accuracy. It increased from 55% (with the RfA method) to 62% (with the RfS method). Nevertheless, the problem is that not only 62% is not a good accuracy, but we can see a significant drop in RfA accuracy too. From 80% to 71%, as shown in Table 4.8. Therefore, RfS is not an effective method of teaching coherency either, since a model that understands coherency should at least be capable of identifying if the last line is randomly selected from a different song.

This suggests that the model may need both the RfA and the RfS methods of selecting negative samples to understand coherency. However, when we created a 50/50 split dataset of RfA and RfS, the accuracy got worse on both labels, as shown in Table 4.8. This led us to test other ways of combining RfA and RfS methods.

Window 4 model	Accuracy on RfA	Accuracy on RfS
RfA training	**80**	55
RfS training	71	**62**
50/50 training	77	56

Table 4.8: Comparing different labeling methods with a four-line window

4.6.1 Combined Training

Looking at the idea of cascade training, we tested a new method we named combined training. It is similar to the cascade training but with a switch between RfA and RfS negative sampling methods in every other epoch.

In Figure 4.12, six epochs of training a four-line window are depicted. Each row is a visualization of only "incoherent" labeled data trained on that epoch. Green squares indicate unchanged lines, and red squares indicate randomly replaced lines. "RfA" and "RfS" in the red squares show the method of selecting random lines.

Like cascade training in Figure 4.10, The first two epochs' "incoherent" labels are randomly generated lines. However, it is RfA generated in Epoch 1, and in Epoch 2, it is RfS generated. In odd epochs, one step is taken to reduce the number of random lines. Note that phases jump from zero "T"s in epoch 1 to two "T"s in epoch three because there is no difference between an entirely random window and a window with only the first line being a "T".

In the last two epochs, we will get two different models. In the last odd epoch (Epoch 5 in Figure 4.12), we get the "RfA combined model", which is trained on the RfA labeled dataset. And in the last even epoch (Epoch 6 in Figure 4.12) we get the "RfS combined model" which is trained on the RfS labeled dataset.

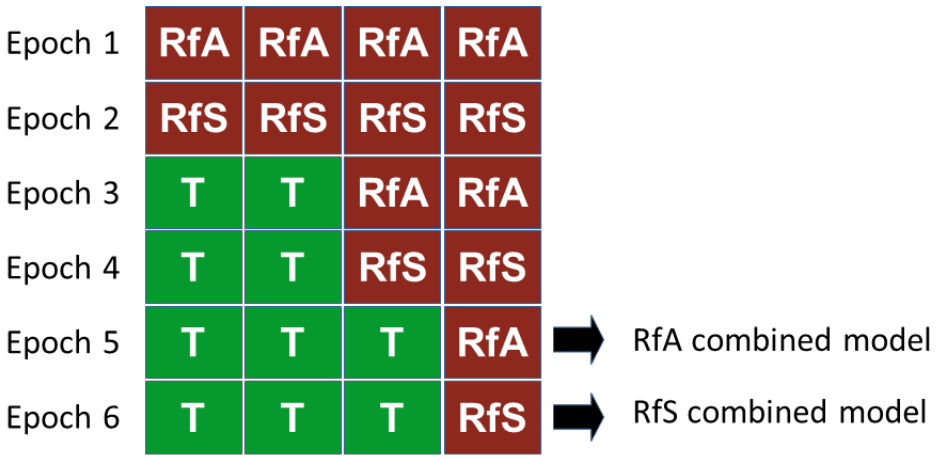


Figure 4.12: Combined training phases in a four-line window

By testing the "RfS combined model", we notice a gain in the accuracy of both RfA and RfS test datasets. As shown in Table 4.9, the accuracy in the RfS dataset increased by 4% from 62% to 66%. This shows that understanding how to distinguish RfA data can also help the model distinguish RfS labels.

However, what surprised us was the accuracy of the "RfA combined model". In our previous experiments, we tried different ways to fine-tune and increase RfA accuracy, and the best we got was 80%, now the "RfA combined model" achieved 81% accuracy on the RfA test dataset. This 1% gain may seem small, but for us, it was an indication that learning to distinguish RfS labels can also help the model get better at distinguishing RfA labels.

Window 4 model	Accuracy on RfA	Accuracy on RfS
RfA training	80	55
RfS training	71	62
50/50 training	77	56
RfA combined model	**81**	56
RfS combined model	73	**66**

Table 4.9: Comparing different labeling methods with a four-line window setting

Understanding the connection between the RfA and the RfS labelings led to a new idea we named micro training in this thesis.

4.6.2 Micro Training

Combined training showed that training RfA and RfS datasets together can help the model get better. Whereas, as we can see in 4.9, training a 50/50 split dataset does not work and can even worsen the accuracy. The micro training method aims to use micro portions of each dataset RfA and RfS to train the model iteratively in each epoch. Each epoch will contain examples of each of the RfS and RfA data. In this thesis, we used 3% of the data

in each epoch. As it is depicted in Figure 4.13, RfA and RfS epochs will be trained in an alternating pattern.

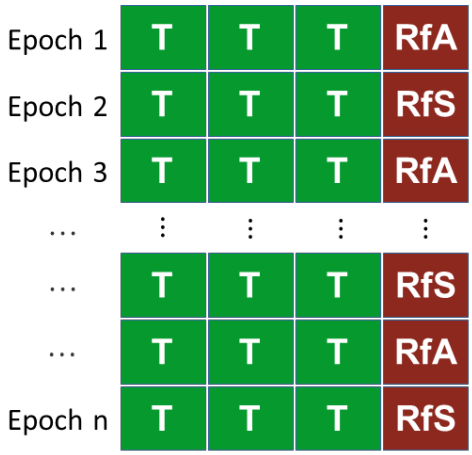


Figure 4.13: Micro training epochs in a four-line window

In both RfA and RfS test datasets, the final model obtained from micro training has the best accuracy. According to Table 4.10, the model did not perform well when trained on a 50/50 split dataset. Nevertheless, with micro-training, the model could learn both RfS and RfA simultaneously. The obtained accuracy for RfS is 67, and for comparison, as can be seen in Table 4.10, it is 5% higher than training with RfS alone.

Window 4 model	Accuracy on RfA	Accuracy on RfS
RfA training	80	55
RfS training	71	62
50/50 training	77	56
RfA combined model	**81**	56
RfS combined model	73	66
Micro training	**81**	**67**

Table 4.10: Comparing different labeling methods with a four-line window setting

When we trained the model with micro training, we evaluated the model's accuracy on both RfA and RfS test datasets in each epoch. The result of this evaluation after training the model for 5000 epochs is shown in Figure 4.14. For a better analysis, we should look closer to see the first 300 epochs in Figure 4.15. With a closer look, we can see that the model accuracy for both RfA and RfS fluctuates in odd and even epochs. For odd numbers, the RfA accuracy is at its high, RfS is at its low, and for even epochs, RfS is at its high and RfA at its low. This is because, in odd numbers, we train a micro epoch on RfA, and in even numbers, we train a micro epoch on RfS data. Nevertheless, as shown in Figure 4.14, this fluctuation will be lower as we train further. The odd and even epochs' accuracy is close to converging in the last epochs.

To better depict how the accuracy changes, we can separate the plotting of odd and even epochs. In Figure 4.16, the accuracy of RfS is depicted with even epochs in blue and odd epochs in orange. As can be seen, the odd epochs accuracy is always below the even

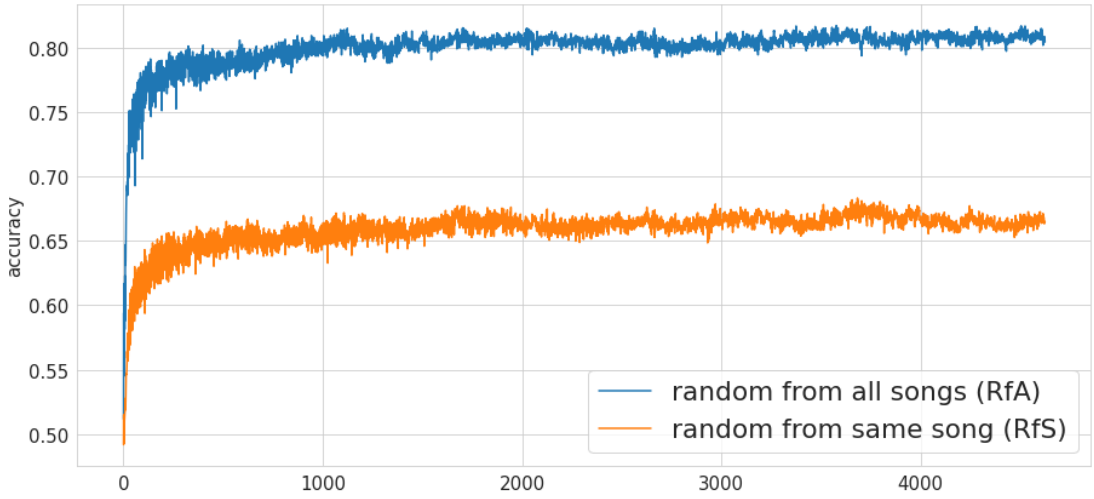


Figure 4.14: RfA and RfS accuracy values in each epoch

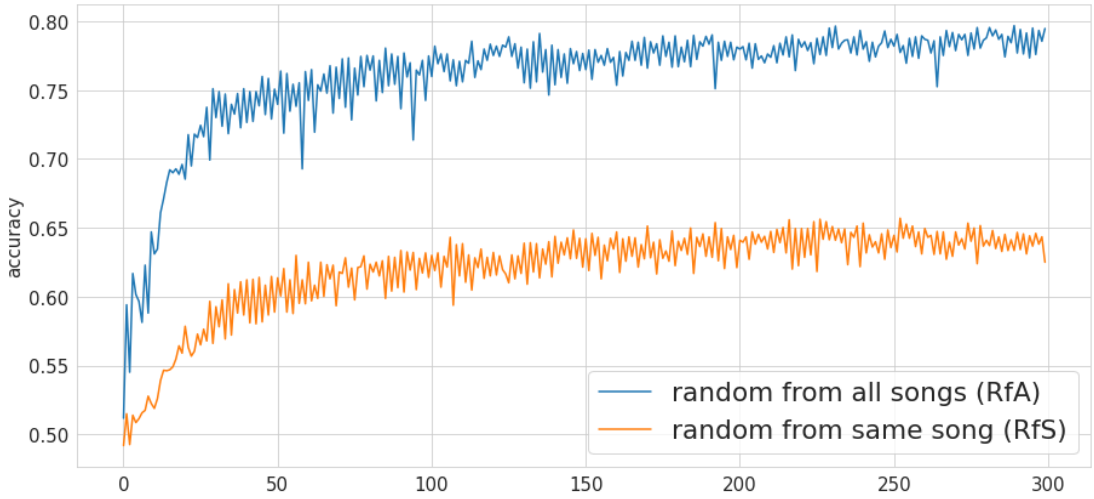


Figure 4.15: RfA and RfS accuracy values in each epoch until epoch 300

epochs accuracy in RfS in Figure 4.16, but as the steps continue, they converge together. At the same time, both of them have a positive trend.

As previously mentioned, one of the problems of RfA labeling was training a model that is too focused on word repetition. In Table 4.6, the score of the RfA model for the Billie Eilish song was negative, and the score for the fake version of it with the last line replaced with only repeated words of previous lines, was +4.4, a very high score. In comparison, this problem seems to be solved for the micro-training model. For example, in Table 4.11, the model's score for Billie Eilish is +1.1, which means it is considered coherent. Moreover, when we tried to trick it by replacing the last line with repeated words from previous lines, the model gave it a -0.05 score which means it did not consider it coherent.

In order to select the initial lines, we need a starter model, as we discussed previously. Using the micro training method, we trained a four-line starter model to achieve an accuracy of 81% on both RfA and RfS start-line datasets. The start-line datasets are a subset of the original datasets consisting of only the start-lines of the poems.

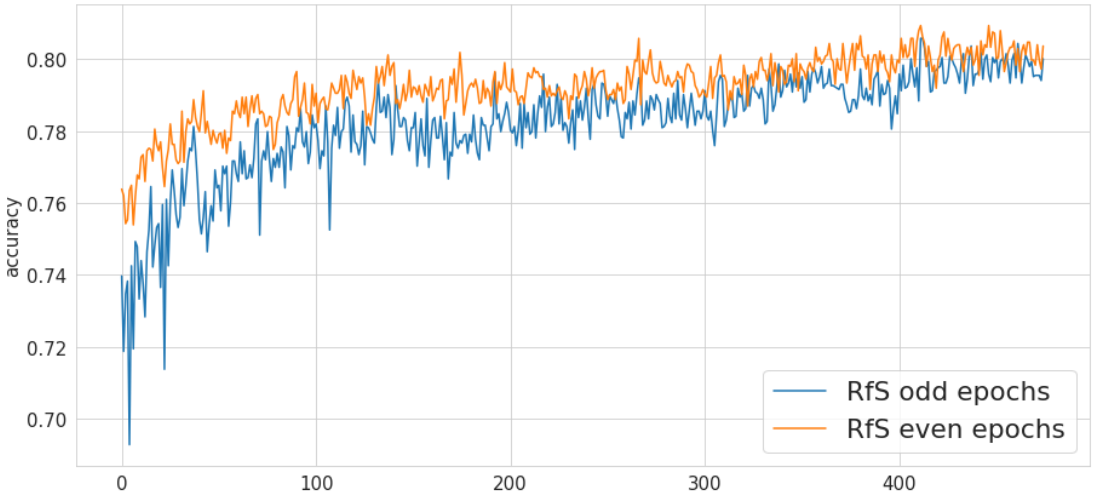


Figure 4.16: Accuracy of RfS in micro training. Even epochs are shown in blue and odd epochs are shown in orange until epoch 1000.

	Song	Score
Real	I should have known [SEP] I'd leave alone [SEP] Just goes to show [SEP] That the blood you bleed is just the blood you owe	+1.1
Fake	I should have known [SEP] I'd leave alone [SEP] Just goes to show [SEP] know leave alone show	-0.05

Table 4.11: RfA cascade training scoring examples.

4.7 Model Evaluation

Coherency cannot be mathematically defined, so we had annotators (UW students) assign coherency labels, which were used to evaluate the model. By placing randomly selected lines from among the lines generated by LyricJam side-by-side with the selected lines from a four-line model, the annotators were asked to determine which poems were more coherent. The order of randomly selected and model-selected lines was randomly assigned in each pair. The annotators were asked to judge coherence based on their common-sense understanding of this concept.

Below are five pairs of poems that were presented to the annotators. The readers are invited to evaluate which poem is more coherent in each pair. The answers can be found immediately following the examples.

Sample 1

when it far away away
hey and never going away
i're getting impossible
they came the answers
free of stone
story that woman's nothing and stay
oh oh let it go away
when you drown away
i'll never be a wrong day
oh i'll be free

the impossible moon
the clouds of his walls
feel impossible inside of me
to leave the answers
here on the wind
fly by my side
lovers fall without you
when you drown away
all the time may disappear
but there's not the only one here

Sample 2

you're going to believe
that i never be sad here
on the moon
and never felt asleep
it's all i know
all i don't know enough
it's not getting out of you
it's time and gone
it's time to change
it's time to see it

i was when the moon
she'll hold my head in me
stay me wild religion
words of what i've lost
they're going to get away
there is no waiting here
when they whispered in the great time
i feel this day
sometimes you think you'll need
you can find and answers

Sample 3

stay with me
fly by my side
there isn't death
i know that's not waiting for me
and i want to go and run
i'mn't chase the one
to the moon again
i will drown away
and then never back to the door
when i'll hide a lover in the moon

feel like fools
oh when it's all you get away
the moon drove to hold the moon
we all we all stream
now he gives all alive
and they're still away
dreams's no chase this one more
the moon is getting
it's never as the sun
i'll hide the answers

Sample 4

when it's hopeless away
all i can still
who will never be alone away
i feel what i'm found
though we should have gone yeah
from spreading from the moon
we were never always time
and fall on by the clouds
to will drown the soul
they're waiting for the one anymore

golden day fools out the sun
it burns inside my own moon
walls come out
and you fall
a million miles away
from all the lights of the hope of you
you're far away
but the moon is a lover
and the place is getting lovers
the time has come

Sample 5

please like the moon
i'll get all our place man
now it's running out this
i'm waiting for the end
life burns the very song of you
oh all the the ocean
always then she's all the moon
keeps us to fall
oh what i'm a song i'm not supposed to stay
to feel your mind

in the land of the clouds
always
even they go away
and when it's still here
i will rain the rain ...
oh to take away the sun
now that i'm waiting
all the world has grown around
all of what's really born
and we are still together

Answers: Poems selected by the coherency evaluator model are Right, Left, Left, Right, and Right. Accordingly, random poems are on the other side.

According to the results of the annotations, the poems composed by the **C**oherency **E**valuator **M**odel (referred to as CEM poems) are more coherent. The CEM poems were labeled as more coherent 78.7% of the time. Figure 4.17 shows the annotation results. Ten out of 30 annotators labeled all the CEM poems as more coherent. It was also found that most annotators (80%) labeled four out of five CEM poems as more coherent than randomly selected lines.

Figure 4.18 shows what percent of annotators labeled the CEM poem in each sample. The sample that has the lowest accuracy is the last one (Sample 5). It is possible that the order in which the samples were annotated contributed to this result.

4.8 Using the Micro trained model

Different methods are available for selecting lines using the model. There has been a discussion of the greedy method earlier in the thesis. There are, however, some other methods that will be discussed in this section.

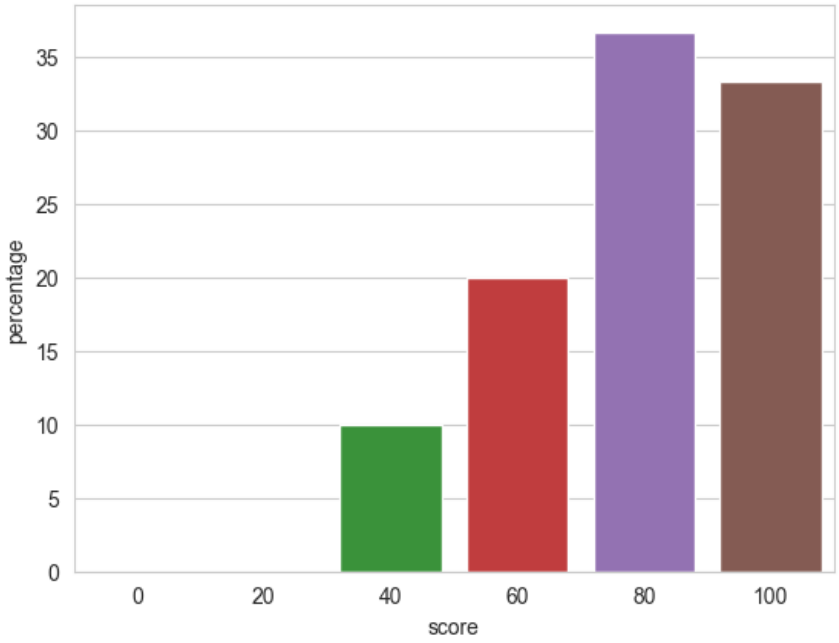


Figure 4.17: Bar plot of preference rate for the selected line poems vs. random line poems. This plot shows how likely a person will achieve a score. For example, 35% of the people answered 4 out of 5 samples correctly, which is 80%, and 10% of the people answered 2 out of 5 correctly, which is 40%,

4.8.1 Different window sizes

The samples used in this thesis are all drawn from a four-line window model. We used the smallest model because the examples have only ten lines. It is possible to increase the size of the window. A bigger window model is found to be slightly more effective in longer poems. We had ten annotators label six pairs of 20-line poems. One of the poems in each pair was composed by a six-line model, while the other by a four-line model. The results are inconclusive, with 55% of six-line model poems selected as more coherent. Further larger-scale annotations are needed to evaluate the effect of the window size on coherence prediction.

4.8.2 Beam Search

The greedy method, however, has one disadvantage: it may select a line that is unsuitable for the next timestep instead of selecting a different line that may provide excellent results for the next timestep. The beam search is another method that can be used to select lines. In beam search, multiple lines are selected in each timestep, and the number of selected lines in each timestep is known as the beam size. The model will calculate the coherency score for the lines in the next timestep based on all of the selected lines before, and again, it will select the beam-size number of the lines and proceed to the next timestep. Figure 4.19 illustrates how beam search works.

Because beam search selects a beam-size number of lines in each timestamp, it is less likely to lose a line that is not the best for the current timestamp but will result in a more

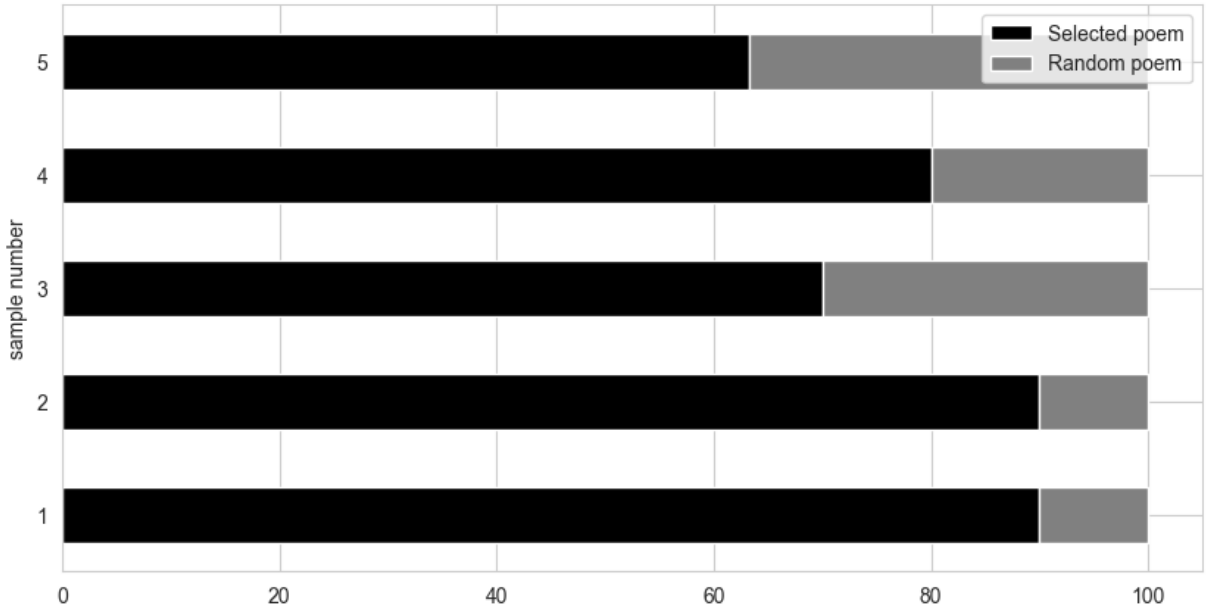


Figure 4.18: Bar plot of rate of preference for the CEM poem in each sample. For example, Sample 4 (4.7) scored 80%, which means 24 out of 30 annotators labeled the CEM poem as more coherent among the two.

coherent poem in the future timesteps, if selected. According to a annotation conducted by eight people, the annotators labeled poems selected using a greedy model over a beam search model with a beam size of 5, 54% of the time. The poems were 20-line long, and we noticed the main focus of annotators in deciding which is the most coherent poem was the first few lines. These results are inconclusive, and a larger-scale annotation is needed to determine whether beam search has any advantages over greedy search.

4.8.3 Mixing scores

It is also possible to mix different models. For example, we can combine the scores of six-line and four-line models to calculate the beam search or greedy method score. The study and evaluation of different methods of mixing models is left for future work.

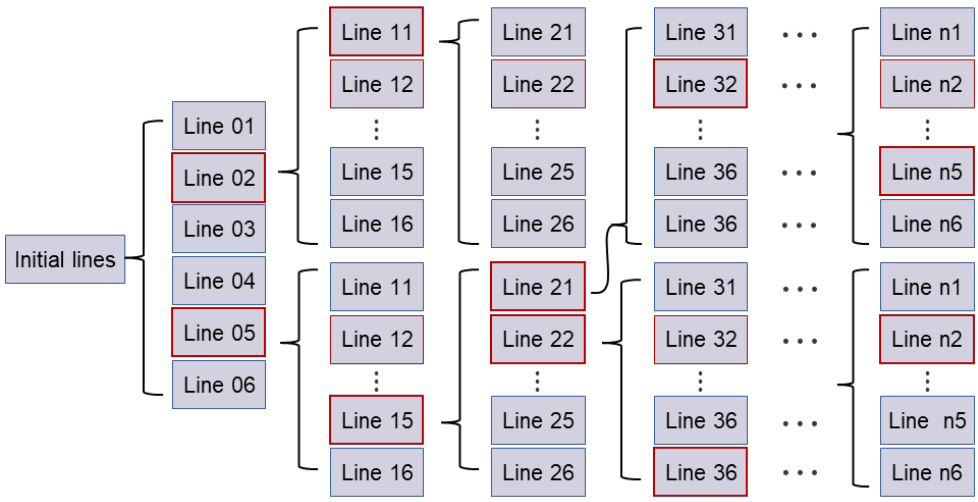


Figure 4.19: Beam search method with the beam size 2

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we propose a model of coherence scoring that allows the system to rank independent lines generated by a VAE and construct coherent poems. However, our method is not restricted to VAEs and can be used with any language model. The scoring model was based on BERT, fine-tuned as a coherence evaluator. We propose a training schedule for fine-tuning BERT, during which we show the system different types of lines as negative examples: lines sampled from the same vs. different poems. To teach coherency to the model, we introduced two methods of labeling, RfA, and RfS. For RfA, the last line in negative (incoherent) samples is sampled from all of the songs in the dataset, and for RfS, the last line in negative samples is sampled from the same song. It has been demonstrated that the model's understanding of coherency is inferior if it is trained only on either RfA, or RfS. We experimented with different techniques to teach coherency to the model, and the best method was determined to be micro-training, where the model is trained by alternating micro epochs with RfS and RfA datasets.
We evaluated the results using annotated data. Thirty annotators were shown pairs of poems in which the lines of one poem were randomly selected while the lines of the other poem were selected by our model. The annotators were asked to label the most coherent poem of the two. The results based on the annotated data, show that poems constructed by the proposed method tend to be more coherent than randomly sampled lines. Specifically, the poems generated by our method are four times more likely to be labeled as more coherent by annotators.

5.2 Future Work

The following areas and research questions would be interesting to investigate further in future research.

First, we introduced two ways of negative sampling, RfS, and RfA, and each of them will teach the model different aspects of coherency. It would be interesting to investigate if there are any other methods of negative sampling to help the model understand coherency better.

Second, The focus of this thesis was only on coherency. It would be interesting to see if this method could be used for stylized poem generation. If we select the positive samples only from one style and include other styles in the negative samples, would this method help generate poems in a specific style?

Additionally, we only used this method for poem generation. It would also be interesting to experiment with this method for long text generation, such as stories.

Finally, it is surprising that the micro-training result is far better than 50/50 split dataset training, as we can see in Table 4.10. If the model is provided with both RfA and RfS labels in the same batch, why is the model not able to learn both? We don't have any mathematical explanation for this, and further research is needed. It would also be interesting to see if we can reproduce the functionality of micro-training in other areas of deep learning such as image classification.

References

[1] Mohamad Abdolahi and Morteza Zahedi. An overview on text coherence methods. In *2016 Eighth International Conference on Information and Knowledge Technology (IKT)*, pages 1–5, 2016.

[2] A. Asperti, D. Evangelista, and E. Loli Piccolomini. A survey on variational autoencoders from a greenai perspective, 2021.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[4] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 02 Jul 2012. PMLR.

[5] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Comput. Linguist.*, 34(1):1–34, mar 2008.

[6] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[7] Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. Discourse-aware neural rewards for coherent text generation, 2018.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[9] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. Evaluation of text generation: A survey, 2020.

[10] Khyathi Raghavi Chandu and Alan W Black. Positioning yourself in the maze of neural text generation: A task-agnostic survey, 2020.

[11] Zhiyu Chen, Harini Eavani, Yinyin Liu, and William Yang Wang. Few-shot NLG with pre-trained language model. *CoRR*, abs/1904.09521, 2019.

[12] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading, 2016.

[13] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[14] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[15] Woon Sang Cho, Pengchuan Zhang, Yizhe Zhang, Xiujun Li, Michel Galley, Chris Brockett, Mengdi Wang, and Jianfeng Gao. Towards coherent and cohesive long-form text generation, 2018.

[16] Woon Sang Cho, Pengchuan Zhang, Yizhe Zhang, Xiujun Li, Michel Galley, Mengdi Wang, and Jianfeng Gao. A bird's-eye view on coherence, and a worm's-eye view on cohesion. *CoRR*, abs/1811.00511, 2018.

[17] K. R. Chowdhary. *Natural Language Processing*, pages 603–649. Springer India, New Delhi, 2020.

[18] K. R. Chowdhary. Natural language processing for word sense disambiguation and information extraction, 2020.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[20] H. P. Edmundson. New methods in automatic extracting. *J. ACM*, 16(2):264–285, apr 1969.

[21] Micha Elsner and Eugene Charniak. Extending the entity grid with entity-specific features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 125–129, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[22] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017.

[23] Alex Graves. Generating sequences with recurrent neural networks, 2013.

[24] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1764–1772, Bejing, China, 22–24 Jun 2014. PMLR.

[25] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013.

[26] Wang Guan, Ivan Smetannikov, and Man Tianxing. Survey on automatic text summarization and transformer models applicability. In *2020 International Conference on Control, Robotics and Intelligent System*, CCRIS 2020, page 176–184, New York, NY, USA, 2020. Association for Computing Machinery.

[27] M.A.K. Halliday and R. Hasan. *Cohesion in English*. English Language Series: A Longman Paperback. Longman, 1976.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997.

[29] Eduard H. Hovy. Planning coherent multisentential text. In *26th Annual Meeting of the Association for Computational Linguistics*, pages 163–169, Buffalo, New York, USA, June 1988. Association for Computational Linguistics.

[30] Keenan Jones, Enes Altuncu, Virginia N. L. Franqueira, Yichao Wang, and Shujun Li. A comprehensive survey of natural language generation advances from the perspective of digital deception, 2022.

[31] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. AMMUS : A survey of transformer-based pretrained models in natural language processing. *CoRR*, abs/2108.05542, 2021.

[32] Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, Austin, Texas, November 2016. Association for Computational Linguistics.

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[34] Donald Knuth. *The TEXbook*. Addison-Wesley, Reading, Massachusetts, 1986.

[35] Solomon Kullback and Richard A Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[36] Philippe Laban, Luke Dai, Lucas Bandarkar, and Marti A. Hearst. Can transformer models measure coherence in text? re-thinking the shuffle test. 2021.

[37] Leslie Lamport. *LATEX — A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.

[38] Joffrey L. Leevy and Taghi M. Khoshgoftaar. A short survey of lstm models for de-identification of medical free text. In *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, pages 117–124, 2020.

[39] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016. Association for Computational Linguistics.

[40] Junyi Li, Siqing Li, Wayne Xin Zhao, Gaole He, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. Knowledge-enhanced personalized review generation with capsule graph neural network. *CoRR*, abs/2010.01480, 2020.

[41] Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. Pretrained language models for text generation: A survey. *CoRR*, abs/2105.10311, 2021.

[42] Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. Pretrained language models for text generation: A survey, 2021.

[43] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[44] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017.

[45] Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. A recursive recurrent neural network for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1491–1500, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[46] Lajanugen Logeswaran, Honglak Lee, and Dragomir Radev. Sentence ordering and coherence modeling using recurrent neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[47] WILLIAM C. MANN and SANDRA A. THOMPSON. Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.

[48] James Martens and Ilya Sutskever. *Training Deep and Recurrent Networks with Hessian-Free Optimization*, pages 479–535. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[49] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531, 2011.

[50] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[51] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics.

[52] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization, 2017.

[53] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization, 2017.

[54] Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1–14, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[55] Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6908–6915, Jul. 2019.

[56] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[57] Ehud Reiter and Robert. Dale. *Building natural language generation systems / Ehud Reiter, Robert Dale*. Cambridge University Press Cambridge, 2000.

[58] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2017.

[59] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online, November 2021. Association for Computational Linguistics.

[60] Tony Robinson, Mike Hochberg, and Steve Renals. The use of recurrent neural networks in continuous speech recognition. 1996.

[61] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, 2014.

[62] Ashley Spindler, James E Geach, and Michael J Smith. AstroVaDEr: astronomical variational deep embedder for unsupervised morphological classification of galaxies and synthetic image generation. *Monthly Notices of the Royal Astronomical Society*, 502(1):985–1007, nov 2020.

[63] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

[64] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[65] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[66] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[67] T.A. van Dijk. *News As Discourse*. Routledge Communication Series. Taylor & Francis, 2013.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[69] Olga Vechtomova, Gaurav Sahu, and Dhruv Kumar. Lyricjam: A system for generating lyrics for live instrumental music. *Proceedings of the 12th International Conference on Computational Creativity (ICCC)*, 2021.

[70] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator, 2014.

[71] Joseph M. Williams and Gregory G. Colomb. *Style: Toward clarity and Grace*. University of Chicago Press, 1995.

[72] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[73] Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. Reference-aware language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1850–1859, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[74] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54(11s):1–38, jan 2022.

[75] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of XiaoIce, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93, 2020.

APPENDICES

Appendix A

Sample Generated Poems

A.1 Poem Samples

Sample 1

here the moon of history
you're lost
when you hear it's mystery
that people that's standing so wild
keeps us far away
but i'm a part of your breath
and we are still together
i feel it's burning
and i feel a lover forget the other side
this time i know

i'll stay away
it's the time to me
i will hear the other
we are a need
assimilation the place of frozen
it means something
the other you are
and impossible moon
you're the other
this time i know

Sample 2

when the sky is gone away
and the moon isn't the still one
i am born of this clouds
assimilation away from the answers
that's already in the sky
i'm the way of this house
i'll stay away from here
oh and just let you feel you feel
all over the end
oh i used to be here

but so peace to me
i was me in my mind
and he's fine
well i see you answers
now everything they sound
i got my mind
the lights that soul
i free brow
i should never really all believe
please

Sample 3

now i don't live
the knowledge of the pain
when he's his in the fire
watching and broken away all broken
never impossible oh
the lights go away
you're born of days
we are all right
that something just a trap here
the things they need the answers
comatose in my heart
i know it's going down
when everything's clear
and i drown still
the walls come away
the lights go away
i can feel this again
i can feel heaven
the moon are slowing on the side
it's all the morning dance

Sample 4

i don't chase out of the moon
all i want to
assimilation in the sky
it's just just like it seems
everything is running out
even now we know
we're lost in this place
lost without a religion
feeling with the hopeless
stay scattered and let go
it's never always of you
and in the glow of the sky line
all of things eyes
but they're all it's in the calm
everything is running out
it isn't believe
i don't believe in here
i am here
then scattered to fall
they're waiting

Sample 5

oh when we're all night
people go away
and try to ...
practiced of the wrong dream
for the story of time
not time again
the moon has lost in the days of you
all the answers of the same
and when we let you place in the sun ...
that's the peace of the morning of you ever
everything i don't know
but it's slowing down
the air are still
it's going to hurt all those things go wrong
and she's still satellite
i don't want to go
here's a killer
now is the answers
the moon moon
i must drive the feeling with time

Sample 6

if i want to know all of time
how this is just a great place
with the great questions
like the earth
with the walls of me
to share the heart
and to see the other end of you
i will find the answers
the scattered questions
drifting out into the space .

the sun without night
hey impossible concentration guitars
oh then i see just a place that's still one
like the earth
stay now
but you feel like later
time the moon .
stay away and it go away away
always scattered
walls hear

Sample 7

i'll stay away
even when i've been enough
a million miles away
from all the lights of the hope of you
now i don't know the answers
but the moon is a lover
and i don't know enough to feel
the time has come
there is no time in my eyes
i can taste the house of mind

now we're still
are are running away
i really don't really like something cry
from spreading from the moon
the answers answers
here in the other you
to will drown the soul
when the mind
she's not cry
my mind

Sample 8

if you're lost
it's my way away
strange moon and dead and clear
keeps me from another soul
there's no more time in the sun
when the world is born
if we want to make this way
fly' out for the night
i can't need this time
when you don't know anymore

all you do
it's fall away
some other time
and everything's a walls
breathe this one more
all my love's born cry
when we're on a bones
the only love still nothing
keeps all the things in the morning
i'll see you all again again

Sample 9

i'll find the moon of the ocean
one of time
here is the sun
the moon is the moon
even then leave another end of the sun
then then happened to believe
he wrote so long away
oh ... you ... still a walls
keeps scattered down in answers
i can't chase this again
one of walls of the moon
one time i was a devil
here at the end
and then i felt still here
oh this place i've never know for more
all that's been down in two
i want to speak on my heart
and when it's still turning
i'll be part of a difference
and when i'm still here

Sample 10

here when the sun
the morning walls to come on
and you're lost
when everything's still
and you can't change the wild side
and when the only love
keeps the world again
keeps away the clouds
the lights of the moon
they let me sleep
oh now i'm still more
the ghosts of what i've waiting
so wild and oh
all the moon .
get all the people
ashes to the river
keeps the world again
i'm the house
let' lord the answers
before i feel like you

Sample 11

you're far away
and have strange need
to go away away
some other place
god can't touch
when he don't touch
people in love
all those who will never feel
is here and see away
if you're broken out of me
i'm waiting and hurt walls
the moon's on
all of love
now i see you
then it was wrong and they
oh you're all in love in you
they're far away
all those who will never feel
i can open away
when the long ignore passed me

Sample 12

it's the question of the morning of miles
and then walk with you
that ship of strange things cry
monsters of moon
i can never see
of everything i need
i wish that i
and the moon is the lover in
a walls they hungry
all this's been going to before
i'm waiting for the moon
the time has come away
and i'm in a saddening touch
with the moon's desires
to create me to the death
of everything i need
but i want the only one
it's waiting for the end of me
one day will be the one one
i'll stay away in the end

Sample 13

i know that's not waiting for me
time to live for
aren't you see ...
oh i don't know anymore here
to the moon
and subtle river of
oh when i take you again
it's made to hurt anymore
i'm not there
and they'll get away the clouds
you got a ride
and i want to go and run
i'mn't chase the one
and impossible moon
when everything's broken out moon
it's the one with nothing with me
i'll hide to make my way
i'm so far away over here
i know everything's lost
and you want this place

Sample 14

all i love is not so much in heaven
others never let me see anymore
everything they've undergone
all the place we'll go
and all the bad time
all the place in the morning past
the sun has never caught this time
and i still have a cry
oh when i'll go out again
i'll stay away from here
it's something to don't know
free on something
and the moon
keeps help me free
and all the bad time
now i see the end again
i'm here
it's getting out of my way
keeps us far
share you'll have passed

Sample 15

to dissolve in
my wild walls of my past
hey and moon and sleep
now never answers is gone .
when it's all the morning way
it's concentration
i will keep it all away
i just want to get enough
playing the moon .
i don't get away
i i want to play this one ...
it's time a time again
and when it's still here
i'll do it all again
it's time to cry out
i'm not to know anymore again
all that matters's running out
this world has nothing to know
i will not go
i'll want to find it all again

Sample 16

when you go away
and you fall away
water vertigo
keeps us far away and
you're still there
and i wish i wish
we were all right away
from the water at the river
leaving all the answers who was wrong
but they're all it's in the calm
is in the answers
but there is still home
water vertigo
the time of what has nothing before
keeps the one to the devil
the sun's a different dream
and you ...
and she's not time
there is all we will find the sun
the sun is still alone

Sample 17

this isn't this time
i am here at the moon moon
the other place
it's my way away
i can't find it here
i can see like this again
how my friend brings you down
strange days they don't cry
i don't hold on
because it's time
into the world that brings you down
when everything go away in here inside
the wrong questions
i will see it all again
then all all that time
and you've like to believe
how my friend brings you down
the strange strange turn on her eyes
here i feel ...
than the moon

Sample 18

summer everything is done
you from the moon
i lost this great look
as we fall on
the lovers they already still look in the mind of god
we don't need a party
that they're never ever get away from
i'm not quite like i feel
stay away
or we all the best forget
they don't get this sound now
like you're mad
and everything is all still one
and the oaks ignore the moon moon
the lovers they already still look in the mind of god
on a moon of mystery
that they're never ever get away from
these summer they're been wrong
stay away
janine janine love should be a ride

Sample 19

i think we've been all before
all i'm still
i don't know anymore
it isn't a long time
i have her cry
summer's time
and they drown
i'm straight free
now everything soul
we now of the moon
summer'm running out
i don't chase this one more time
there's a young love in the way
wandered scattered by the sun
she's in the only dream yeah
dressed in love in heaven
and impossible in the end
and in the moon in the morning
is the world in the sunlight
and we've felt fine

Sample 20

it's time and get away
with the sleep with my eyes
the light moon
even though they're just too long away
i'll hide away the answers
i'll listen to you ...
everything is everywhere i see
even impossible's in love with heaven
hey now never let you know the lovers
i'm all the lover in the way
the wild love and come
wisdom eyes i need
this place born to play way in my soul
you can choose the answers
oh people return
is a fool of you ...
everything is everywhere i see
are you from the clouds again
wandered used to forget
i know things's going to change

Sample 21

that only you born in here kiss
you will never be the end
we're going to fall
oh and touch
the lights by the moon
even then we see alone
with all i'm a here
god now i'm asleep
the lights and guitars still look
oh my heart

everything that's strange
keeps it away
keeps scattered by the other moon
strange tears and very wild eyes
when you've lost the other moon
it will be away in the earth
though everything is gone
forgive the land of a young moon
the clouds of my moon
it never was not so clear here

Sample 22

when you drown
you can tell the answers
but all the wrong time
they're far away
hey the answers now
all i need to know anymore
the program has this time to see
the ghosts that's always before
while the sun and the devil
and then the chase love to roam

now till things the answers
she says the waiting is end
the lights touch you can fly
but you're away a little
i am free
when you've been a great place
you don't chase
they're rain to the end
while the sun and the devil
and impossible

Sample 23

it's far away
here the moon's over
we're lost in this place
lost cells the thoughts of days
want to fall to come to me
i need this place
i can't let it go wrong
it's time i look down
i don't chase a sound
and i won't lose the strain

so came from this moon are not the answers here
here the moon
we're lost in this place
assimilation in one more days
i am lost to believe
i'm out of you
and offer me in
that's getting getting part of my heart
the moon moon
before they've made it all wrong

Sample 24

and the moon
watching us happened
then wild time twice
and i'm the things you've
when you want me
is the ones in the moon
but i'm happy
that burns that gets us on
when i'll get away
while the moon is the lover moon

when everything is running out
i like to mind
everything all at once
it's waiting for the run and the mind
lost like a glare in a ride
watching you fall
but it's still alive
it burns so hard to believe in
all that i see
is everything i do

Sample 25

the sun is waiting away
i'm all there's here
watching you drown
i'll give the morning one time
all the moon
and they's waiting by heaven
they give her a morning of a world
like we moon
never'll get away all
and tell me the world and heart is shame

watching the moon
when you get away
watching you drown
and dissolve in the
monsters of my religion
coercion and hold on the clouds
my wild walls of my past
when i want the end
i'll do it all again
it's time to cry out

Sample 26

i by the clouds
something in my dreams i was wrong
the knowledge used to forget
soon the body says i lose my friend
and they give up hope and all
dead in the end of nothing
the impossible is impossible
i'm just a killer of a day
the morning of places i can't take
tomorrow's the poison of the door

and other place
they wrong to be free
all i've
this is never time
oh you give me
stay impossible
all impossible money
and when a killer on a own
i'm running out
television with the heart

Sample 27

it seems concentration
that you realise
the lights and the moon still
the wind and the moon of the night
all the arms in the sun
of all the places of their eyes
that the world on the moon
all this world is still one
and the land is nothing clear
there's all i believe
it's me what you believe in me
today oh when i's nothing
now it's a end
breath the streets of the moon moon
is us just in a great reality
it's not over
you go away
she was ragged and our place
but it's you
to create a place of a great monster

Sample 28

while when they are still to run games
you can't find all
here the moon
this time
strange time they live again
i can feel this place of you
than the moon
a million mistake
when you don't know anymore
then had time and back in the waves
watching the moon
strange moon and dead and clear
keeps me from another soul
this is a dead time denied
there is no love
when we feel like two
fly' out for the night
keeps all the things in the morning
keeps nothing nothing will find the answers
i'll never learn

Sample 29

i feel the clouds in the moon
they're born to fade away
i can't let them fall
with all i see
all the light of history
i want to get all of love
everything in me
i can't know it all
the open things inside
they're not right
please
and then i come waiting
all i see
all you've
the lights of the moon
all of oaks
and all one word time
easy long sun and a long dream
you are born to answers asleep
and i'm running out

Sample 30

people give me to hide
and when i get away
i'm gone
i don't know what's right
with all i need
i'll do it all again
it's time to cry out
please
all that matters's running out
this world has nothing to know
and dissolve in the
some don't want to play the other side
it's land of you ...
and then i was not enough
i have lost everything
god will let all the answers
now you're young now ?
the clouds of me
they're born to ride
walls us far away

Sample 31

everything can justify you
all of mystery
here's the moon still
i will let you fall
and you fall away
long long long dream free
and they fall
it's time to look anymore
now we want to make the part of day
it's already to get in the moon
my heart is to the answers
all of mystery
i'm never place this one more
into the arms of this great nothing mind
the moon think i will look in the eye
then i'll get free free free
the moon is a lover
keeps you still
the sun's a different dream
than of the other days

Sample 32

when i'm all dead ...
i seek for chase still more
he's an end
when you want ... in all morning ...
keep the things that's true
and now we will
it's what you suffer
and the moon still
the lights moon
to keep your eyes and never give me ...
it's concentration
it's far away
on all of what i feel
i give in my way to believe
one day is not the same one
and the morning ignore the other one
i want to find the one man
who laid in the moon
now i want to know ...
oh the sun is gone and the sun

Sample 33

i'm far away
all that you used to roam
there's no place to go free
i like the lover in the sun
away away from the light
drifting into the sun .
the light of the heart of manhattan
the woman in my heart
oh the moon in the money's time
here is the moon
now you want to know that you ...
what used to roam
forgive the answers
all everything i need
the morning isn't let the earth in
i want a satellite
oh to let it go asleep in
of us fools ...
with the moon is more than a one
keeps this see the end

Sample 34

there isn't death
everything is going to die in all over you
time to live for
oh you don't know what it's
the wrong place
stay i drown
and subtle river of
stay the moon
is gone ? ...
i'm waiting for the other one
when i see the moon
i know that's not waiting for me
and i want to go and run
i'mn't chase the one
to the moon again
i will drown away
and then never back to the door
when i'll hide a lover in the moon
the morning moon is dead and the other side
i'm waiting for the other one

Sample 35

oh the moon .
i couldn't have it all again
we're lost
the fire of the devil
now if i'm waiting
hey all we need
she's not right
hut you fed
it's bankrupt away
now this difference of this great time
i don't know the moment
walls came tumbling in
a impossible concentration
like the silence
it's waiting
now you've felt the one
that you only get
all the water of the song
left to sleep with the wall
now this difference of this great time

Sample 36

strange tears of me cry
and i drown away
even then i feel
though everything is gone
god now i'm asleep
even so far away
with the moon is nothing to love with me
sometimes the moon
i will drown
i will drown away

then they ride down the fly
all the land of the moon
we're staying right with you
trail of my soul
when we'll still in the moon
hey the moon
i want to speak to believe
and all in all is slowing over ...
you willn't find the moon and i know that i know
the sun will never found the sun

Sample 37

i'm the ocean night
and the moon is falling
the other moon
i know it's coming
the moon has spoken
i'll be born in the moon .
oh with the house that's mine
i don't let it fall
oh the lord of the moon's all the way
i'm the lover in the sun

my love is nothing to feel
forgive us out of this great more time
all we will make this
oh there is a burning out of time
it isn't a long time
or at her lies
we were all we're only allowed
here is not for the moon
oh ... let you get away
this is never there in the one you

Sample 38

when this place they need the end
it's concentration
and i want a mind
it's not a part of the day
and there's a great soul
when you want her in satan .
i fall the answers
the other moon and frozen's people
one over time
they're hopeless

i used to get away
when everything is running out
now everything in the mind
all i've been seem to make
i can't chase the soul
when he's like a lover in a road
it's time to make this way
it's not go to nothing again
the lovers they've never look so far away
the ghosts has a lover in the moon

Sample 39

make the things goal
in return of you
on people go away
to make the answers
when it's all the earth
and if it's all inside of me
i admit it's here
you're in the past
the program i think i'm coming
and i will protect
i'm never felt clear here
when everything is more
the walls will let the end
and then take a little miles again
when i think i will still know you
let the walls go tumbling in
i will drown
a monument by you
oh when we're only one
i will hide in this place of time

Sample 40

i can drown you
all this time
i'm free ocean
and keep the dead man
i don't chase someone
i want the more in the earth
i listen to my mind
i'll hover away the sky
you know i can't stay
it means something i had to know
watching you in the light
the lights turn of my mind
all those who came to forget
all they want to go
they're born of the moon
they're not like they know you feel
oh the sun is gone and the sun
the clouds of the moon of the moon
oh all the general dream
you're all the answers of all

A.2 Labels

The labels for the above samples are listed below; 0 means the poem lines are randomly selected and 1 means poem lines are selected by our model:

Sample number	First poem	Second poem
1	1	0
2	1	0
3	0	1
4	1	0
5	1	0
6	1	0
7	1	0
8	1	0
9	0	1
10	1	0
11	1	0
12	0	1
13	0	1
14	1	0
15	0	1
16	1	0
17	1	0
18	0	1
19	0	1
20	1	0
21	0	1
22	1	0
23	1	0
24	0	1
25	0	1
26	1	0
27	1	0
28	0	1
29	1	0
30	1	0
31	0	1
32	0	1
33	1	0
34	0	1
35	0	1
36	1	0
37	1	0
38	0	1
39	0	1
40	0	1