# Attention to Lyrics

Ken Trinh, Emily Fernandes, Joseph Ritter

University of California, Berkeley
iy3827@berkeley.edu, emily.r.fernandes@berkeley.edu, joe.ritter@berkeley.edu

December 1, 2022

**Abstract**

*We advance lyric generation research by enhancing existing models with a self-attention mechanism and by investigating the coherence and perceived authenticity of machine generated lyrics. We explore lyric generation using both recurrent neural networks (RNN) and transformers across three models: (1) RNN using Long Short-Term memory (LSTM) cells; (2) RNN using LSTM cells with multi-headed self-attention; and, (3) a GPT transformer fine-tuned on specific genres. We first determine if the genre-specific generated lyrics are representative of the targeted style by using a BERT-trained categorization model. Then, with a semantic graph algorithm, we measure the generated lyrics coherenceness. Finally, we test the authenticity of the machine generated lyrics by surveying a human interpretation of the lyrics.*

## I. Introduction

Music lyrics sit at the intersection of semantics, metonymy, and interoperability, resulting in a rich field for NLP research. For example, lyrical style is variable across time and within genres, and what constitutes 'good' lyrics will always be subjective. Lyrics can elicit a wide range of emotions and memories for some, and are easily ignored and forgotten by others. These effects are idiosyncratic with one song potentially having millions of meanings within a single audience.

Given the fungible nature of song lyrics, our work explores the usefulness of NLP models, trained in genre specific corpuses of text, with the goal of generating realistic lyrics, across eight different music genres. This research potentially has commercial viability as musicians, producers, and other music-industry stakeholders often directly outsource lyric generation to song writers or collaborate with them. This body of work has the potential to assist songwriters and the users of their lyrics as an enhancement to or supplement for their labor.

This article is organized as follows: we begin with a brief summary of existing research followed by an outline of our methods and models, provide an analysis of the results, and conclude with ideas for extending this research and future commercial applications.

## II. Related Works

The greater body of work in lyric generation research has been promising but, as far as we can tell, singularly focused in either music genre or the type of neural network or language model employed. For example, Potash et. al (2015) [4] focused their lyric generation within just the rap genre and at the individual artist level. Watanabe et. al (2018) [9] focused on a broader set of music genres, but were limited to 1,000 Japanese songs. Gill et. al (2020) [1] in introduced a wider array of genres to expand the aforementioned research but focused on just one model. Recently, Wang et. al (2021) [8] used lyrics from just seventy-five artists to generate lyrics from a given song's title. While Rodrigues et. al (2022) [6] used a fine-tuned GPT-2 model to study the model's effectiveness in overcoming challenges with metaphors, metonymy, syntax, and semantics.

## III. Methods: Data Processing

The data was originally scrapped from Genius by Gill et al. [1] across eight genres of music[1], The scraped data are highly unbalanced, with Rock having the most samples (10,022 songs) and Folk having the least amount of samples (3,983 songs).

[1]Target genres: Folk, Jazz, Metal, Pop, Rap, Rock, R&B, and Soul

**Table 1:** *Lyric Genre Sample*

| Genre | Sample Size | Training Size |
|---|---|---|
| Folk | 3,983 | 3,000 |
| Jazz | 4,840 | 4,000 |
| Metal | 7,833 | 6,000 |
| Pop | 8,248 | 6,000 |
| RnB | 4,335 | 4,000 |
| Rap | 9,957 | 6,000 |
| Rock | 10,022 | 6,000 |
| Soul | 4,064 | 4,000 |

### i. Data Sampling

Since we construct lyric generation models independently of each other (more below), we down sample the Metal, Pop, Rap, and Rock genres to 6,000 samples each. The Folk, Jazz, R&B, and Soul genres were capped to 3,000, 4,000, 4,000, and 4,000, respectively, because these four genres do not have more than 6,000 samples. We avoid uniform down-sampling every genre to 3,000 samples to ensure the models access as much information as possible within their respected lyric genre. Table 1 provides a breakdown of the lyric generation training set.

### ii. Data Pre-Processing

Within each genre, we apply a series of pre-processing steps. First, each lyric is cleansed for a set of stop words. For the RNN models, we keep the special new line token since, purely from an aesthetic perspective, lyrical structure usually breaks into a new line without properly ending a sentence. We also keep some of the punctuation [“<!>”, “<?>”], once again for style purposes, and then tokenize all of the lyrics. From the tokenization of each lyric within the genre, we create a unique set of tokens to serve as the genre-specific vocabulary. This vocabulary is used to set up two dictionaries converting word tokens to integers and vice versa.

We use a decoder-based architecture to train a lyric generating model which introduces additional data pre-processing requirements. First, within the same genre, the length of a lyric is not standardized. Second, the input data needs to be structured in a way that the model can take a representation of a prior output to use as an input to predict the next word. To meet this requirements, we cap the input sequence length to 16 words and use the $17^{th}$ token at each line of input as the target. The $17^{th}$ token is then encapsulated into the next set of $16^{th}$ tokens. Figure 8 (Appendix) describes our input-to-output data structure.

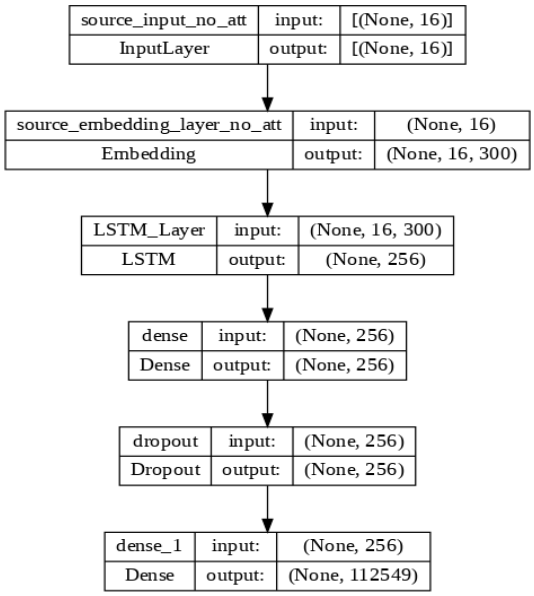


**Figure 1:** *Baseline RNN with LSTM cells*

## IV. Methods: Text Generation Models

We compare three types of models. First, as a baseline, we utilize RNN architecture using long-short term memory (LSTM) cells [10]. Second, on top of the baseline RNN [7], we will apply a self-attention layer on the LSTM output. Finally, we fine tune a GPT-2 language model [6], with a decoder based transformer on the structured data set.

### i. LSTM without attention

As mentioned earlier, we deploy the same RNN architecture with LSTM cells [10] as Gill et. al (2020) [1]. This is our baseline model and functions as a point of comparison for additional features. In the provided literature, the network consisted of 5 layers: embedding layers, LSTM layers with dropout, a dense hidden layer with non-linear a activation function, a dropout layer, and a dense layer with a softmax activation function. Figure 1 (above) outlines the architecture for the baseline LSTM model.

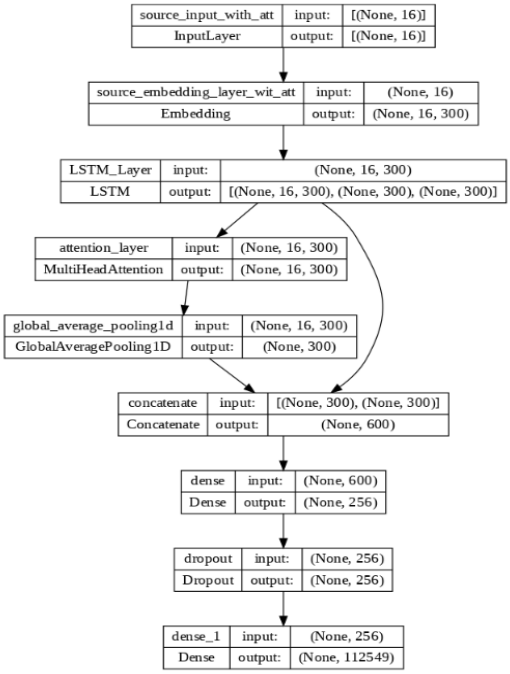


**Figure 2:** *RNN with LSTM cells and Multi-Headed Attention*

### ii. LSTM with attention

We test the hypothesis that adding attention to the proposed LSTM architecture [10] is, in fact, 'all we need'. We update the simple LSTM architecture by adding a multi-headed self-attention layer to the LSTM output. Within each genre, we apply self-attention using a Tensorflow's Multi-Headed Attention implementation to the output of the LSTM layer. For this implementation, we use 10 different heads with a query and key dimension of 5 to avoid overfitting and to reduce the computational complexity of computing a large dimension attention head. Figure 2 describes the new architecture.

### iii. Transformer Model

Recent advances in NLP has lead to the development of transformers. For our research, we use a decoder-based transformer, GPT-2 [6]. As previously noted, a decoder-based transformer fits our lyrical generation task because a decoder transformer would self-attend to previously generated words. Since GPT-2 was trained on a large corpus of data, we fine tune the model independently to all eights genres.

For each genre fine-tuning task, we cap each lyric set in the training sample to 1,024 tokens and remove all stop words in the given list: [ " <newline> ", " <tab> ", " # ", " ' ", " ( ", " ) ", " ; ", " : ", " - ", " [ ", " ] " ]. A final <|endoftext|> is appended at the end of each of the 1,024 token samples. Each token is embedded using the pre-trained GPT-2 word embedding. The data set is fed into the GPT-2 model as an input and as an output for this fine-tuning process.

### iv. Genre Categorization Model

In order to evaluate the generated lyrics' genre, we train a multi-class categorization model off of the corpus of lyrics used to train our genre lyric generating algorithms. This categorization model is trained off of 80% balance data to have identical examples of each genre. Lyrics are transformed using a trainable BERT model (maximum word count of 100). The <CLS> token from BERT is passed to a neural net which contains a dense layer (dimensions of 301), then a 30% drop out layer in order to mitigate overfitting, culminating in a softmax layer with the length of the number of lyric genres. Refer to Figure 9 (Appendix) for the architecture of our categorization model.

## V. Scoring Metrics

### i. Lyrics Genre Classification

We train our lyric generation model independently on genre and hypothesize that the generated lyrics will reflect their native genre. Therefore our classification model is constructed to classify whether or not the generated lyric falls into the target genre. The classification model uses BERT as an encoding layer and a feed forward neural network for classification. Further detail on the classification model follows.

### ii. Coherence Semantic Graph

Given the abstract nature of generating lyrics it is challenging to quantify and measure success. Using a BLEU or ROUGE score could provide metrics to score our model, and these metrics are traditionally associated with other NLP tasks; but, in our case, the scores are insufficient. For example, the BLEU score [3] is commonly used to evaluate machine translation between two languages. The text generation task in this study generates lyrics solely in English, rendering the BLEU of little to no value. Likewise, the ROUGE score [2] is commonly used to evaluate a model's recall ability in summarizing tasks. Again, we find this metric provides little to no value for our task.

Therefore, in this study, we use three semantic graph algorithms, preceding adjacent vertex (PAV), single similar vertex (SSV), and multiple similar vertex (MSV) proposed by Putra et al. [5]. Each of the three methods were first introduced to determine the coherence of all sentences in the input text. We believe coherence is an ideal indication for text generation tasks because the generated text should appear authentic–common flow, sentence structure and patterns. In the literature [5], each sentence of a given text is first tokenized, a word embedding is applied, and the result is input into the algorithm as nodes.

All three semantic graph algorithms use cosine similarity between the word embedding of a reference sentence and the word embedding of another sentence within the same text as a weight measurement. A graph is then set up with all the sentences as nodes and the weights as edges. Upon the construction of the graph, equation 1 is then use to compute the text coherence [5]:

$$tc = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{L_i}\sum_{k=1}^{L_i} weight(e_{ik}) \quad (1)$$

Where N is the number of sentences in the text and L is the number of outgoing edges from the vertex.

### iii. Survey Based Authenticity Measure

Aside from these metrics, we conduct multiple surveys to test whether or not the machine-generated lyrics are distinguishable from the human-written lyrics. For this experiment, we create a survey for each genre, and each survey contains three lyrics written by humans, three lyrics generated by the baseline RNN, and three lyrics generated by our RNN with attention. Each lyric contains three choices: (1) human written; (2) machine generated; (3) I know this song. The addition of "I know this song" controls for survey takers who already know the lyrics in question.

## VI. Results

Evaluating the effectiveness of our lyric generating algorithm is accomplished using the aforementioned three metrics. We think of these as content, syntax, and human judgment results. For content, we use our categorization model to validate if the generated lyrics are true to the targeted genre. Secondly, we evaluate the generated lyric's syntactical correctness by using the semantic similarity metrics: PAV, SSV and MSV.

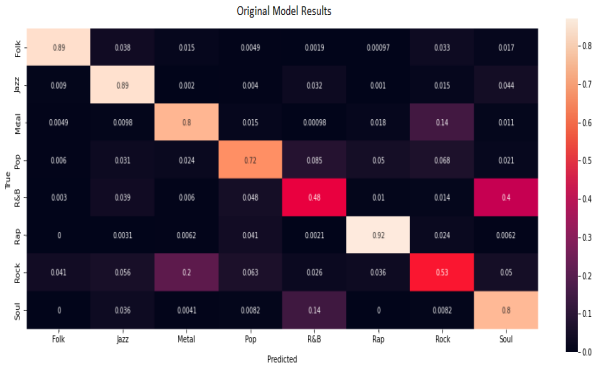


**Figure 3:** *Confusion Matrix for categorization model evaluating on the 20% testing data. (Note: only the rows representing the True responses will sum to 1)*

**Table 2:** *Categorization Model Genre Specific Accuracy Score*

| Genre | Human | Basline | Attention | GPT |
|---|---|---|---|---|
| Folk | 0.89 | 0.45 | 0.43 | 0.26 |
| Jazz | 0.89 | 0.38 | 0.45 | 0.51 |
| Metal | 0.80 | 0.48 | 0.36 | 0.25 |
| Pop | 0.72 | 0.24 | 0.26 | 0.16 |
| R&B | 0.48 | 0.11 | 0.10 | 0.06 |
| Rap | 0.92 | 0.68 | 0.47 | 0.80 |
| Rock | 0.53 | 0.29 | 0.28 | 0.49 |
| Soul | 0.80 | 0.79 | 0.87 | 0.36 |

These graph-based metrics capture the relationship of the words within the text. Lastly, we evaluate human judgment via surveys. While the first 2 metrics attempt to capture the generated lyric's syntactical correctness and content, we believe human judgment is an important, paramount variable.

### i. Content Metric: Categorization Model

To serve as a point of comparison, we use the remaining 20% of the balance data set on the categorization model to compute a confusion matrix (Figure 3). Then we generate lyrics for each of the three models using the first 16 words from a random sample of 100 lyrics per genre. The generated lyrics are then fed into the categorization model. The confusion matrix for each model, across all genres, is reported in Figure 10 (Appendix). Table 2 summarizes the accuracy scores across our test set, baseline, LSTM-attention, and GPT-2 generated lyrics (1 = perfect accuracy).

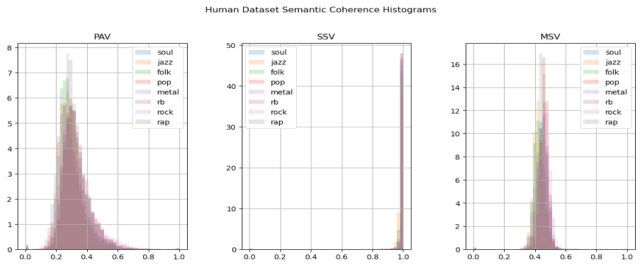


**Figure 4:** *Coherence Score For Original Training Data Set*

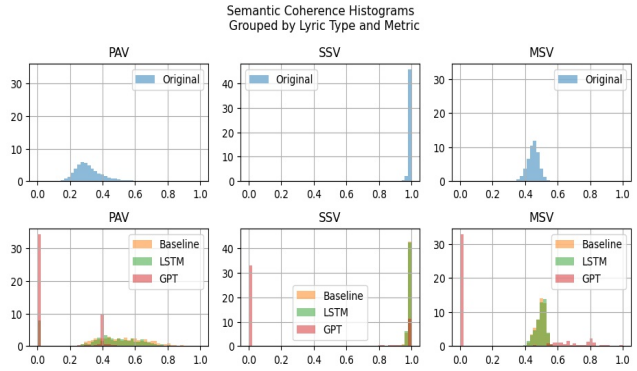


**Figure 5:** *Original vs. Models' Coherence Score Distributions*

## ii. Syntax Metric: Semantic Similarity Graph

To further assess the coherence of the generated lyrics, we use our coherence score for the generated lyrics. In order to understand how our generated lyrics measure up against human written lyrics, we first explore how these metrics look for human written lyrics.

In Figure 4 (above), the semantic metrics for lyrics written by humans for all genres fall within similar patterns. PAV for all genres is centered around 0.3 with a slight right skew. SSV for all genres closely bunch between 0.8 and 1.0 with a left skew. MSV looks normally distributed and centered around 0.45. With this in mind, we calculate the same semantic metrics for baseline, LSTM attention, and the GPT-2 generated lyrics. From the human written data set, we observe that while there is some variation per genre, there is a definite general shape of the genres' semantic metrics. Figure 5 (above) showcases the coherence score for each of the models.

## iii. Human Judgement: Survey

To properly capture a human's interpretation of the generated lyrics, we randomly sample USA-based adults to determine if they can detect which lyrics were

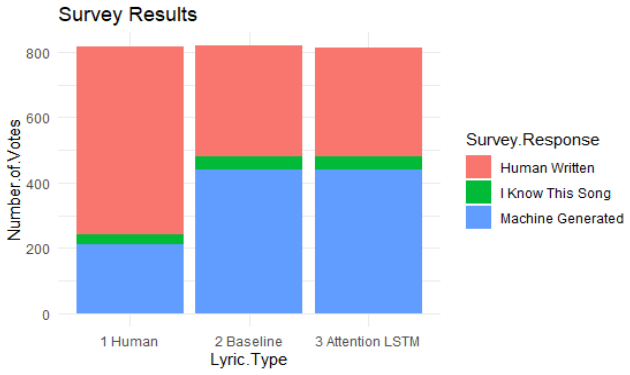


**Figure 6:** *Survey Responses for Human vs Machine Generated*

written by humans and which were written by our models. 9 lyrics were randomly sampled from each genre and were proportionally distributed between each lyric generation type: human written, baseline model, and attention LSTM model. From this random sampling we create 8 surveys, one for each genre[2]. A total 30-40 responses for each survey question and the surveys followed the above mentioned structure. Figure 6 reports the results of the conducted surveys.

It is clear that the computer generated lyrics can be detected by human judgment. However, to quantify this relationship, we used R to predict the surveys' 'Machine Written' votes and percent share of the surveys' 'Machine Written' votes using the lyrics generation type as input. This method resulted in the table reported in Figure 7 (next page).

With that, we compile a summary of the means and confidence intervals for raw survey responses and percent share survey responses broken out by lyric generation type and survey response category that can be found in Figure 11 (Appendix).

# VII. Discussion

## i. Content Metric

Our results show that five of the eight genres have an accuracy over 80% with only two categories, R&B and Rock, performing less than 50%. Interestingly, our categorization model shows a tendency to categorize Rock as Metal and R&B as Soul approximately 14% of the time. This could be due to a number of factors such as the similar usage of slang, the ways verses were

[2]SurveyMonkey was used to create and host the surveys which were published through Positly, who charged a small fee for finding and vetting survey participants.

| | Dependent variable: | |
|---|---|---|
| | Machine.Generated Machine Generated Votes | Machine.Generated.Percent Machine Generated Percent |
| | (1) | (2) |
| Lyric.Type2 Baseline | 9.583*** | 27.867*** |
| | (1.719) | (4.931) |
| Lyric.Type3 Attention LSTM | 9.625*** | 28.471*** |
| | (1.979) | (5.781) |
| Constant | 8.750*** | 25.762*** |
| | (1.270) | (3.777) |
| Observations | 72 | 72 |
| R2 | 0.337 | 0.346 |
| Adjusted R2 | 0.318 | 0.327 |
| Residual Std. Error (df = 69) | 6.487 | 18.647 |
| F Statistic (df = 2; 69) | 17.537*** | 18.263*** |
| Note: | | *p<0.1; **p<0.05; ***p<0.01 |

**Figure 7:** *Survey Statistical Analysis*

constructed, or factors not captured in lyrics alone (for example the instruments or beats associated with each genre).

We believe that our classification model over or under predicts some genres since those genres do not sum to 1. In this analysis, machine-generated lyrics resulted in a significantly lower accuracy score than human-written lyrics across all genres. As expected, all machine-generated lyrics continue to confuse Metal for Rock, R&B for Soul, and vice versa. Interestingly, the lyrics generated by the baseline model tend to over predict the genres of Jazz, Rock, and Soul while those generated by the LSTM with attention model tend to over predict the Soul genre. GPT-2-generated lyrics fare the worst with an over prediction in Jazz, Rap, Rock, and Soul. We reason this phenomenon is rooted from the data being fed into each genre's model requiring additional cleansing or the machine-generated lyrics can not distinguish between the lyrical lexicon across all of the presented genres. The genre classification metrics can be found in Figure 10 (Appendix).

### ii. Syntax Metric

In our syntax analysis, Figure 5, we reported that the baseline and LSTM with attention models generate lyrics with very similar semantic metrics. The SSV and MSV metrics look almost identical to the human written lyrics. The PAV results of machine-generated lyrics increase slightly toward 1, from a range of 0-1, with a wider range of higher scoring values. This suggests that lyrics generated by the machine is somehow more coherent than those written by a human(s). Upon inspection of the generated lyrics, we observe that the lyrics generated by both the baseline and the attention models have few repeated lines. Since we opted for a high alpha value (0.6) in favor of similar unique overlapping terms, this could be the result for the positive deviation.

The GPT-2 model results were vastly different than the human generated. After visual inspection of the generated lyrics it is easy to see why. Many of the generated lyrics were repetitive, connecting ideas and concepts that were unnatural, and containing inexplicable symbols. Since both the content and syntax metrics for GPT-2 result in large deviation from the original lyrics, we exclude the GPT-2 generated lyrics from the human study.

### iii. Human Judgement

Figure 7, column 1 'Machine Generated Votes' shows that the average votes for 'Machine Generated' when the given lyrics are human written is 8.75 votes. When lyrics were generated by our baseline model, humans voted 'Machine Generated' an additional 9.583 votes for a total of 18.333. When lyrics were generated by our LSTM with attention model, humans voted 'Machine Generated' an additional 9.625 votes for a total of 18.375.

Column 2, 'Machine Generated Percent', highlights that the average percentage of the survey responses for 'Machine Generated' for human written lyrics was 25.76%. When lyrics were generated by our baseline model, humans voted 'Machine Generated' an additional 27.87% for a total of 53.63%. When lyrics were generated by our LSTM with attention model, humans voted 'Machine Generated' an additional 28.47% for a total of 54.23%.

Both columns report statistically significant results. This suggests our survey results do not fall out of random variation. Unfortunately due to limited resources we can not determine statistically meaningful results within each genre as we only had three questions which were answered approximately 30 to 40 times for each Genre/Lyric Generation Type group. However, we do have plots broken out by genre in the appendices to fully show the variations between genres.

## VIII. Conclusion

This study shows that lyrics generated by LSTM-attention models were categorized in its respected genre better than that of the baseline model and GPT-2 model. The LSTM-attention model over-predicted on one genre compared to three genres by the baseline model and five by GPT-2. Our coherence metrics show that the GPT-2-generated lyrics where vastly different

than that of the baseline, LSTM-attention, and original lyrics. Interestingly, both baseline and LSTM-attention lyrics contain less repeated line than that of human-written lyrics, which contributed to a higher coherence score. Finally, our survey results show that regardless of which model was used to generate lyrics, other people can still pick out the differences between human and machine. While that result is somewhat disappointing, it does highlight areas of improvement to our proposed LSTM-attention model for future study. Furthermore, incremental advancements in this research will likely result in commercial viability for more advanced models. It is not difficult to imagine how these models will be used to complement ideation or as a supplement to the creative process of writing lyrics.

## References


[1] Nick Marwell Harrison Gill, Daniel (Taesoo) Lee. Deep learning in musical lyric generation: An lstm-based approach. *Yale Undergraduate Research Journal*, 1(1), 2020.

[2] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[3] 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, 2002.

[4] Peter Potash, Alexey Romanov, and Anna Rumshisky. GhostWriter: Using an LSTM for automatic rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1924, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[5] Jan Wira Gotama Putra and Takenobu Tokunaga. Evaluating text coherence based on semantic similarity graph. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pages 76–85, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[6] Matheus Augusto Rodrigues, Alcione Oliveira, Alexandra Moreira, and Maurilio Possi. Lyrics generation supported by pre-trained models. 35, May 2022.

[7] 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.

[8] Yuyan Wang, Jason Chuen, Eunji Lee, and Katherine Wu. Popnet : Evaluating the use of lstms and gpt-2 for generating pop lyrics from song titles. 2021.

[9] Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. A melody-conditioned lyrics language model. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 163–172, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[10] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, 31(7):1235–1270, 07 2019.

```
sentence_1 = ["there's", "two",
              "things", "\n",
              "that", "i",
              "have", "yet",
              "to", "learn", "\n",
              "how", "to", "forget",
              "or", "have"]
target_sentence_1 = ["i"]

sentence_2 = ["have", "i",
              "love", "someone", "\n",
              "all", "these",
              "time", "waiting", "\n",
              "have", "come",
              "suddenly", "when",
              "did", "i"]
Target_sentence_2 = ["feel"]
```

**Figure 8:** *Input Features & Output Target Architecture*

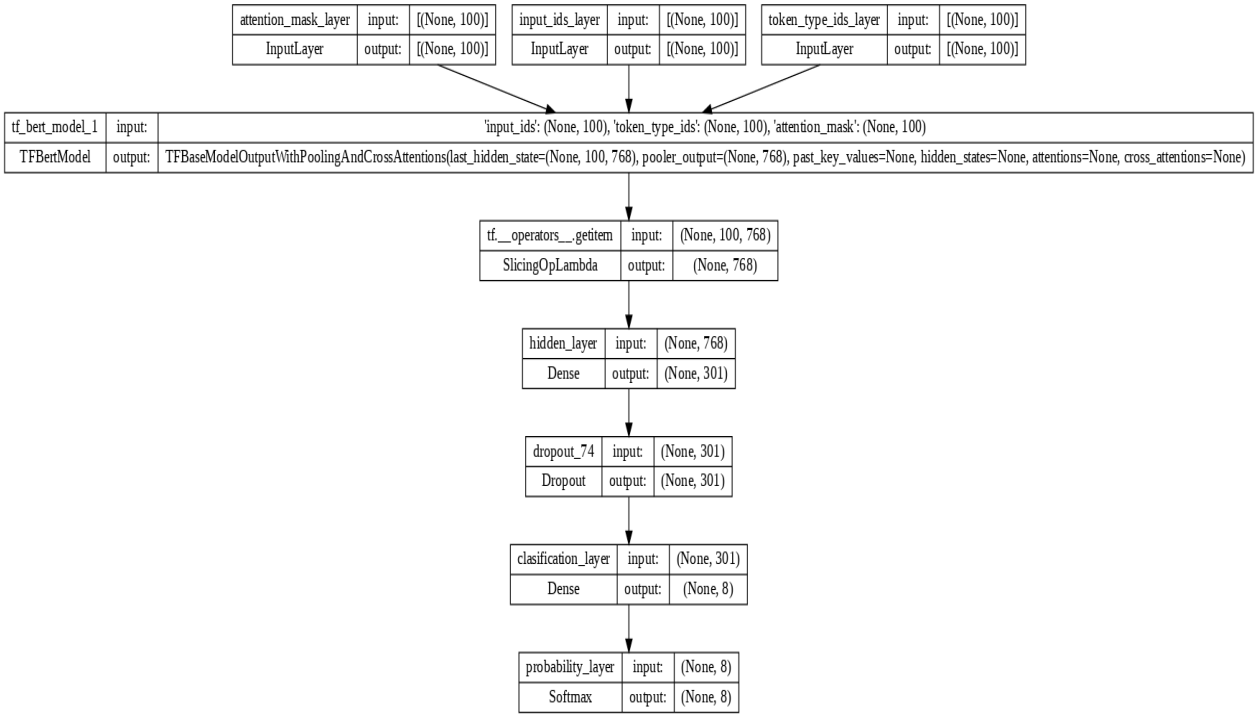


**Figure 9:** *Genre Categorization Neural Network with LSTM cells*

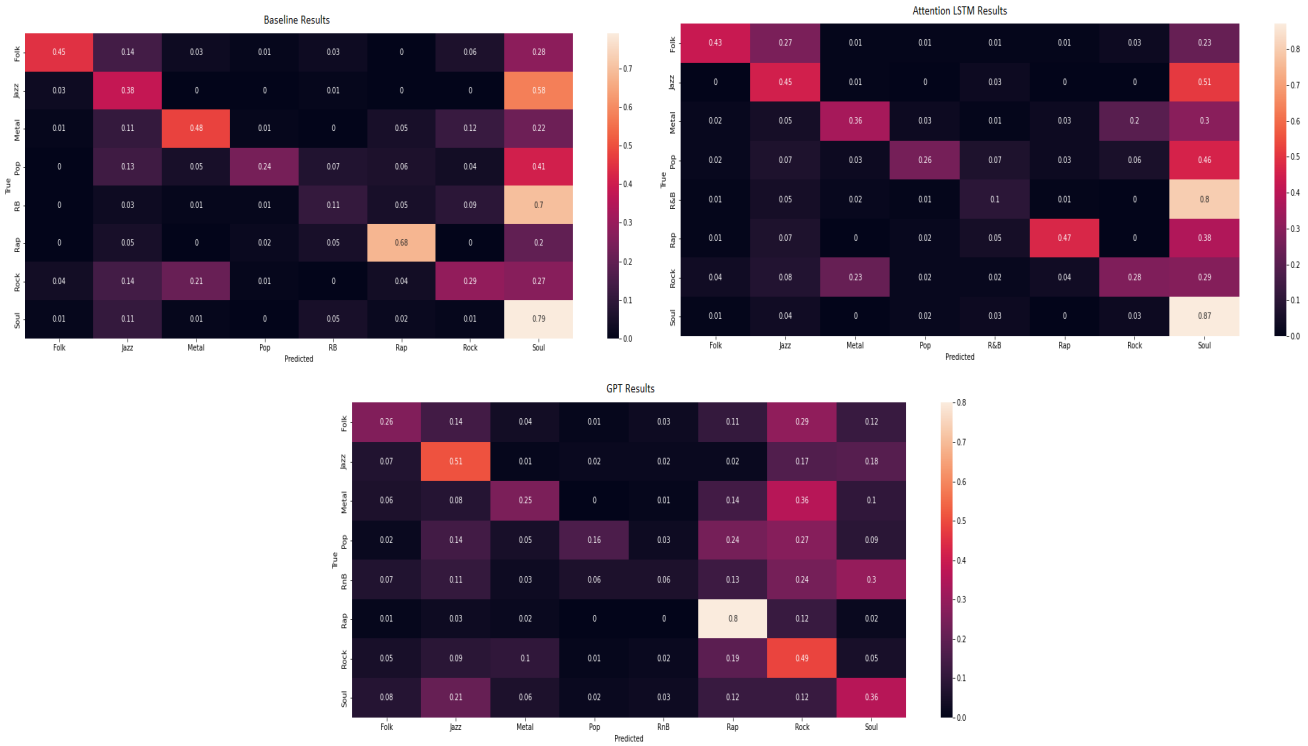


**Figure 10:** *Confusion Matrix for categorization model evaluating on the basline model (top left), lstm-attention model (top right), and GPT-2 (bottom). (Note: only rows representing True responses will sum to 1)*

| Survey Groups | | Survey Resonsponces | | Survey Percentage | |
|---|---|---|---|---|---|
| Lyric Generation Type | Survey Response | Mean | Confidence Interval | Mean | Confidence Interval |
| Human | Human Written | 23.9 | (21.1, 26.6) | 70.3 | (62.4, 78.2) |
| Human | Machine Generated | 8.8 | (6.1, 11.4) | 25.8 | (17.9, 33.6) |
| Baseline | Human Written | 14.1 | (11.9, 16.3) | 41.6 | (35.0, 48.3) |
| Baseline | Machine Generated | 18.3 | (15.9, 20.7) | 53.6 | (47.1, 60.2) |
| Attention LSTM | Human Written | 13.9 | (10.8, 17.0) | 41 | (32.1, 49.9) |
| Attention LSTM | Machine Generated | 18.4 | (15.2, 21.5) | 54.2 | (45.2, 63.3) |

**Figure 11:** *Survey Statistical Analysis: Summary of Confident*

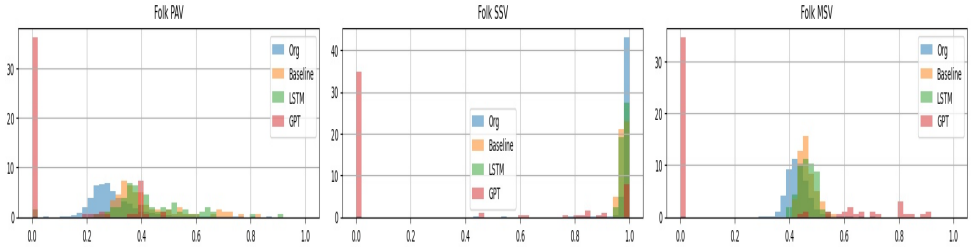


**Figure 12:** *Folk Semantic Distribution*

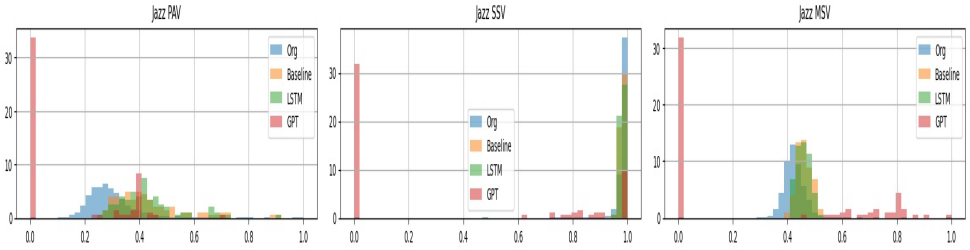


**Figure 13:** *jazz Semantic Distribution*

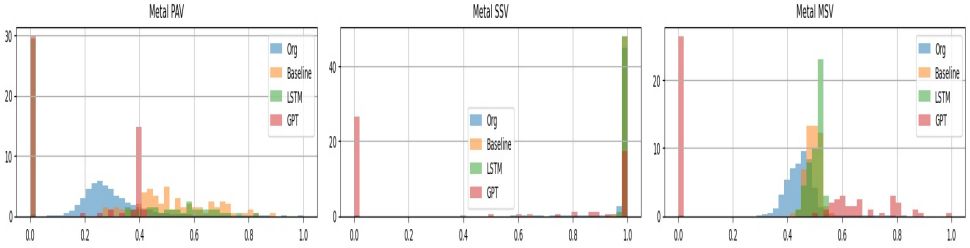


**Figure 14:** *Metal Semantic Distribution*

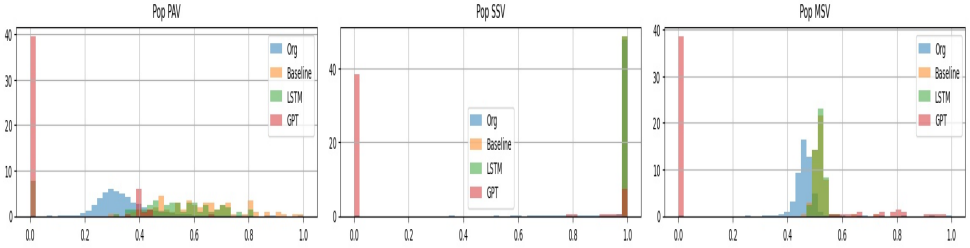


**Figure 15:** *Pop Semantic Distribution*

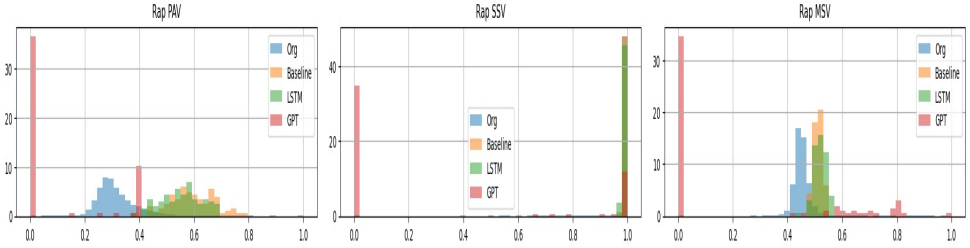


**Figure 16:** *Rap Semantic Distribution*

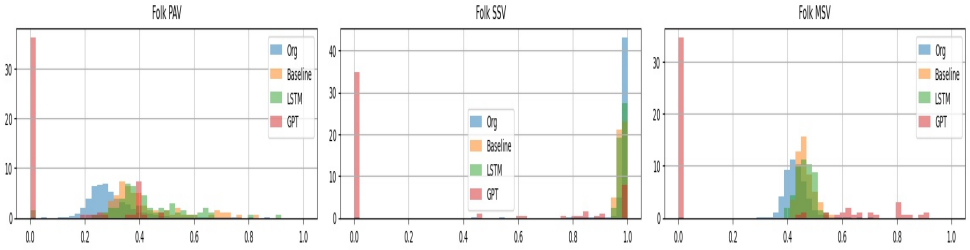


**Figure 17:** *Folk Semantic Distribution*

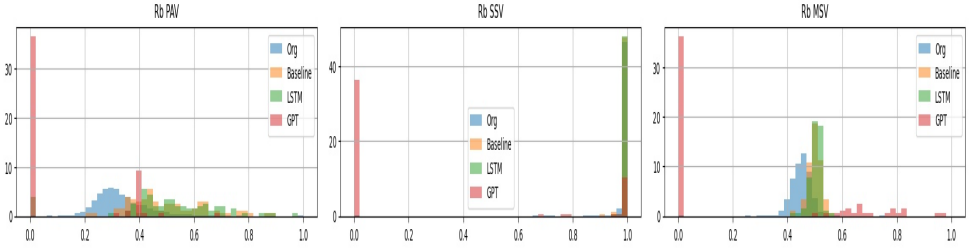


**Figure 18:** *R&B Semantic Distribution*

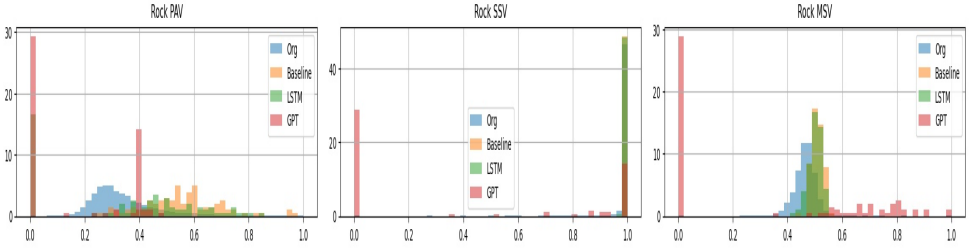


**Figure 19:** *Rock Semantic Distribution*

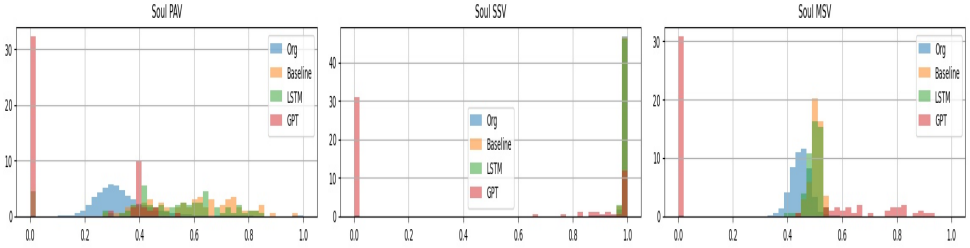


**Figure 20:** *Soul Semantic Distribution*


**Figure 21:** *Folk Survey Results*

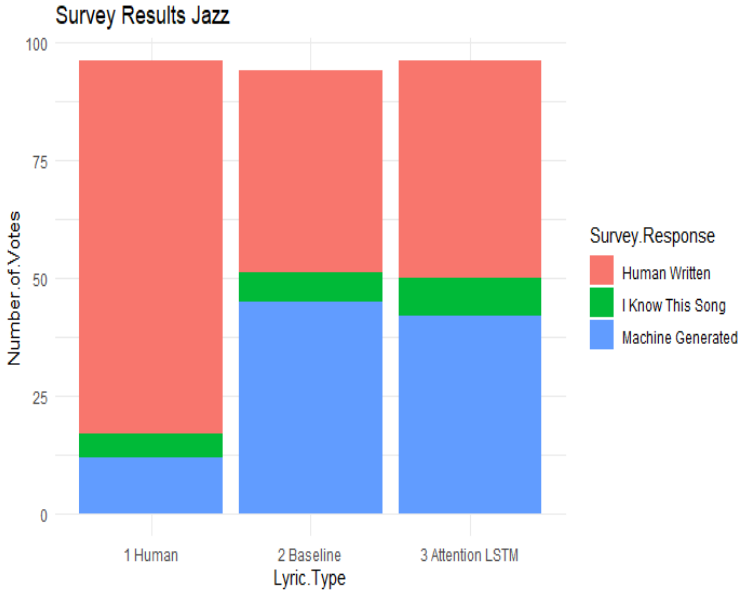


**Figure 22:** *Jazz Survey Results*

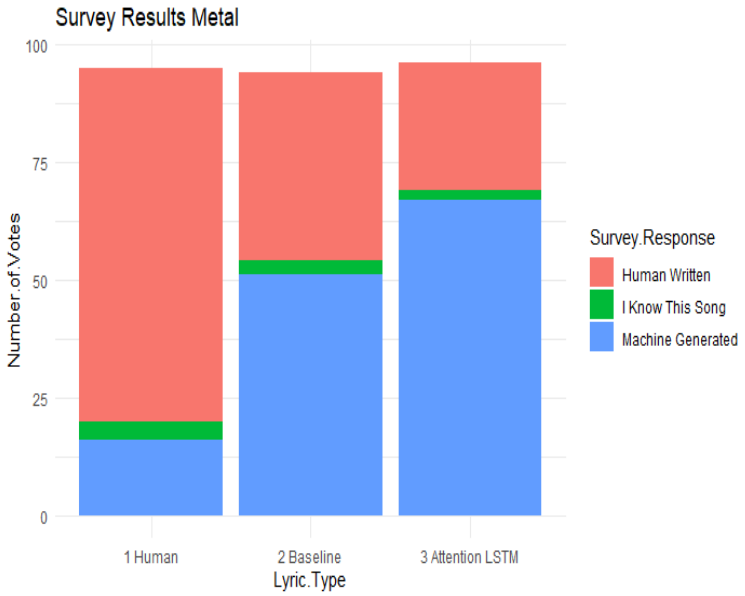


**Figure 23:** *Metal Survey Results*

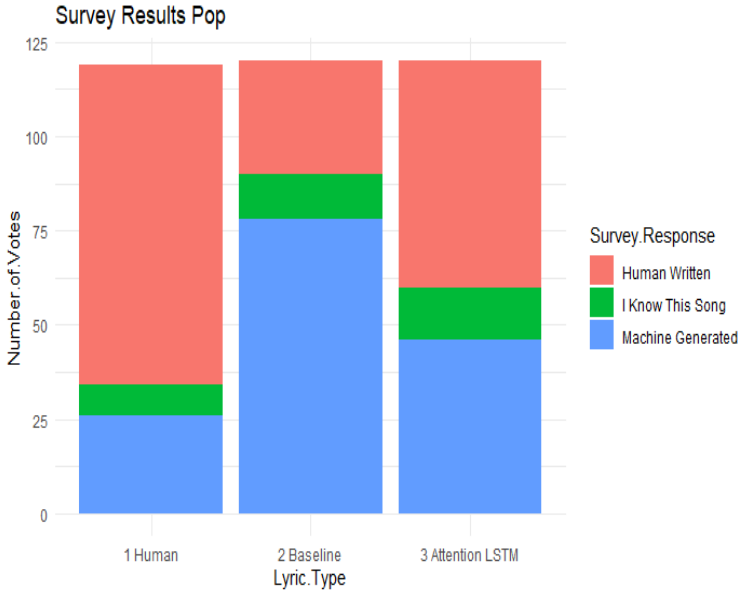


**Figure 24:** *Pop Survey Results*

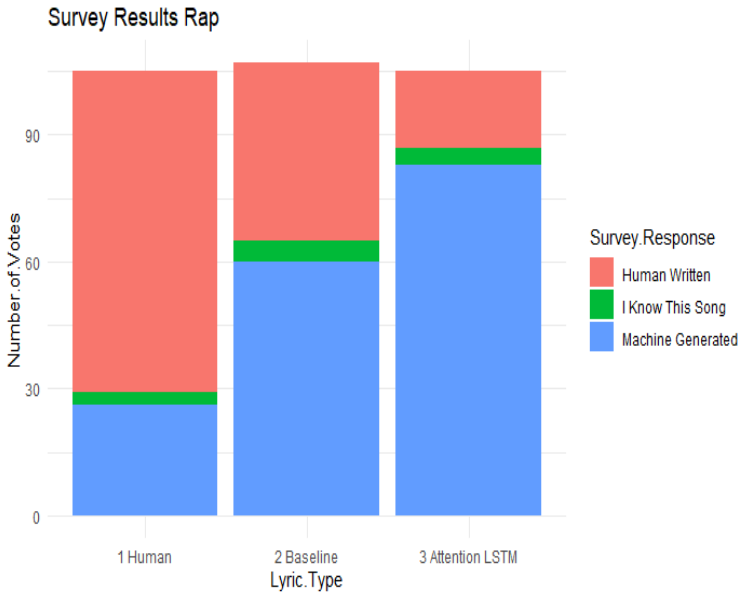


**Figure 25:** *Rap Survey Results*

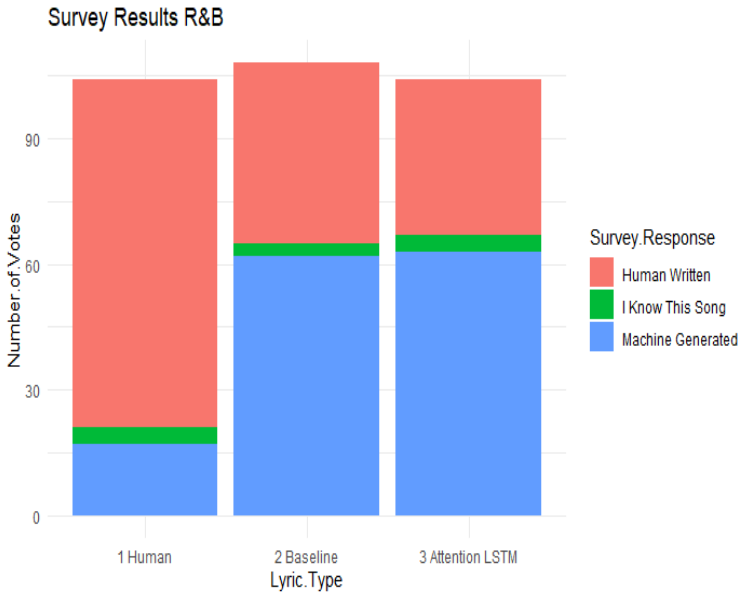


**Figure 26:** *R&B Survey Results*

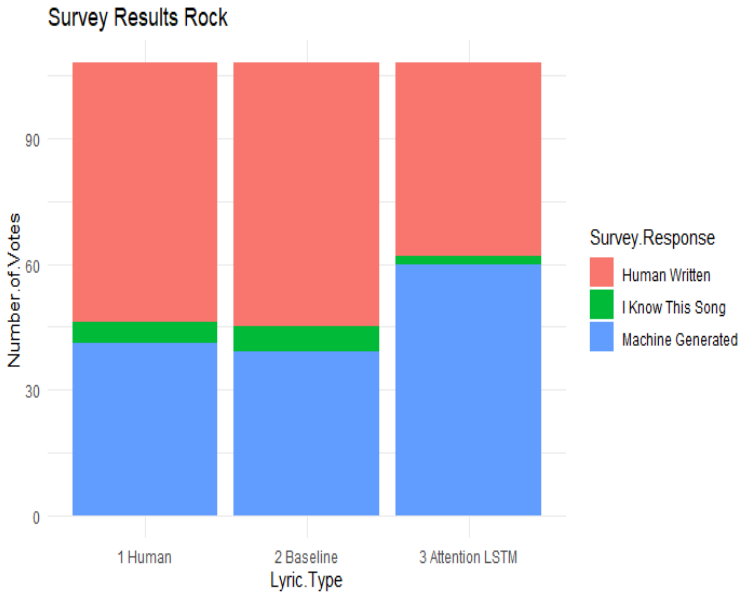


**Figure 27:** *Rock Survey Results*

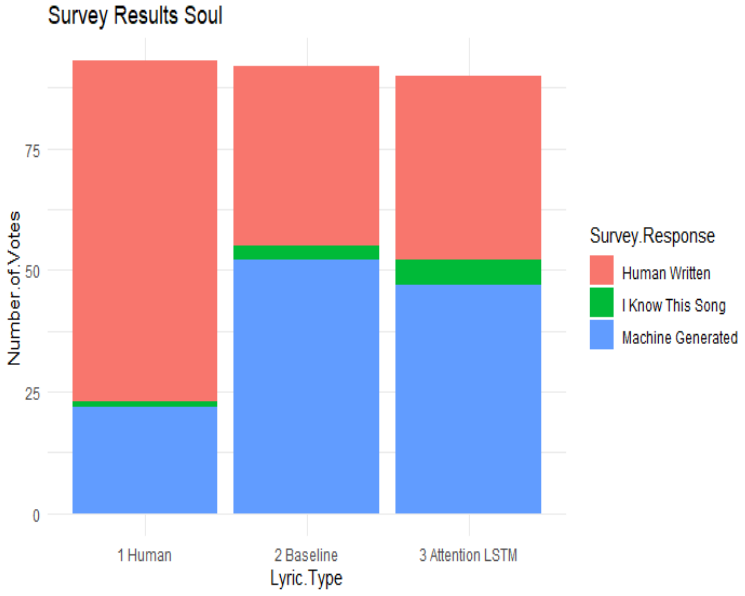


**Figure 28:** *Soul Survey Results*