Taylor-Made Lyrics: Keyword-based Lyric Generation

Aditi Baskar
abaskar@umass.edu

Akshaya Mohan
akshayamohan@umass.edu

Niel Parekh
nieljiteshpa@umass.edu

Suraj Jain
surajjain@umass.edu

1 Problem statement

The automation of lyric generation has garnered significant attention from researchers in recent years. Numerous lyric generation tools have been developed, enabling users to create complete songs based on inputted words. These tools serve as valuable resources for artists experiencing writer's block and seeking inspiration for new ideas.

The quality of generated lyrics has notably improved due to advancements in neural networks, particularly in text generation. Previous studies have primarily focused on specific music genres, such as pop and rap, employing deep learning models to capture the genre's underlying semantics and style.

In this work, our objective is to explore lyric generation based on user-provided keywords while adopting the distinctive style of a single artist, Taylor Swift. By inputting keywords that represent the desired theme of the song, our proposed model aims to generate lyrics that successfully encompass both Taylor Swift's writing style and the user's intended theme.

To achieve this, we curate a dataset by collecting Taylor Swift's song lyrics and employ the GPT 3.5 model to extract keywords from each song. Subsequently, we fine-tune a Flan-T5-Small model specifically for the task of lyric generation. To evaluate the performance of our model, we compare it against a baseline n-gram model. The evaluation focuses on assessing the preservation of the artist's style and the meaningful representation of the provided keywords within the generated lyrics.

By undertaking this work, we aim to explore the field of lyric generation by creating a model that combines the unique lyrical style of Taylor Swift with user-specified themes.

2 What you proposed vs. what you accomplished

- Dataset compilation:
  - Scrape lyrics from the Genius website using the Python client LyricsGenius
  - Extract keywords from each song using ~~KeyBERT~~ → GPT 3.5 (on further research and discussion with the TA)
- Lyric-generation task: Fine-tune a T5 model → used FLAN-T5-Small due to computational restrictions
- Baseline model: n-gram model
- Evaluation: Stylistic and semantic evaluation

3 Related work

In recent years, significant progress has been made in the field of lyric generation, with a focus on rap lyrics due to its standard structure and stylistic and semantic features such as rhyme and rhythm. For instance, Malmi et al. (2015) utilize an information retrieval approach to generate unique verses line-by-line, by ranking candidate lines from a repository using the RankSVM algorithm. Nikolov et al. (2020) introduces a transformer-based denoising autoencoder that is trained to reconstruct rap verses based on extracted keywords from existing lyrics, allowing for the generation of novel verses. LSTM-based models have also been commonly used for rap lyric generation, with Manjavacas et al. (2019) comparing different approaches such as character-level, syllable-level, and hierarchical language models based on templates focused on rhythm and rhyme. GhostWriter (Potash et al., 2015) proposes an LSTM-based system that models the style of a target artist to generate unique lyrics similar to their existing work, with a focus on both style and song structure.

The application of LSTMs extends beyond rap lyrics. Gill et al. (2020) employ an LSTM to generate lyrics for specific genres, tested on the pop genre as well. Melody-lyric pairings have also been explored, as seen in Watanabe et al. (2018), where an RNN-based model generates lyrics based on given melodies. These works primarily aimed to replicate the style of a particular genre.

To generate lyrics in the style of a single artist, Vechtomova et al. (2018) combine audio and text modalities using variational autoencoders, making inroads in the realm of multimodal lyric generation. Transformer models have also been utilized, as demonstrated by Ram et al. (2021), which was the first to employ the T5 model to learn lyrical and stylistic features for pop lyrics generation. Xue et al. (2021) introduce DeepRapper, a rap generation system that utilizes a Transformer-based autoregressive language model to model rhymes and rhythms comprehensively. This system incorporates vocal and lyric alignment, as well as lyric and beat alignment into the results.

Our approach begins by addressing the task of keyword extraction from lyrics text. Notable contributions in this area include the work of Nikzad-Khasmakhi et al. (2021), which proposes a multimodal key-phrase extraction approach. This approach treats the task as a sequence-labeling problem and utilizes transformers and graph embeddings. The performance of this method is evaluated against BERT and ExEm. Another relevant study by Qian et al. (2021) combines key-sentence extraction using BERT with various methods such as TF-IDF, text rank, and LDA to extract keywords from text.

Evaluation metrics for generated lyrics encompass various aspects. Text-level metrics, covered in Gill et al. (2020), Nikolov et al. (2020), Ram et al. (2021), Malmi et al. (2015), include word repetition/diversity, line length, and rhyme metrics like rhyme fluency, rhyme accuracy (Xue et al., 2021) and rhyme density. Stylistic evaluation is approached differently, such as training a CNN classifier to classify songs of an artist for evaluating style transfer (Vechtomova et al., 2018). Briakou et al. (2021) explores and evaluates style transfer metrics, suggesting alternative approaches like modeling it as a regression task to resemble human evaluation.

4 Your dataset

Our dataset consists of 319 Taylor Swift song lyrics, and 3-5 keywords corresponding to the main themes of the song.

4.1 Lyrics

The lyrics utilized in this study were obtained from Genius.com (`https://genius.com`), a prominent American digital media company. Genius.com hosts a user-driven platform where individuals can contribute annotations and interpretations to various forms of content, including song lyrics, news articles, poetry, and documents. To access the song lyrics data stored on Genius.com, we employed the Python library LyricsGenius (Miller, 2017, 2020). This library offers a convenient interface to retrieve information related to songs, artists, and lyrics from the Genius.com database. Leveraging the capabilities of LyricsGenius, we successfully downloaded the complete discography of Taylor Swift for our analysis.

4.1.1 Data Pre-Processing

We utilized the LyricsGenius library to implement specific filters that excluded Remixes and Live versions of songs. This was essential to avoid any potential repetition in the dataset. It should be noted that Taylor Swift has re-recorded and released two of her previous albums. To ensure uniqueness, we retained only one version of the re-recorded songs.

Furthermore, we encountered instances where the downloaded songs included content other than actual songs, such as awards acceptance speeches or album commentaries. To ensure the purity of the dataset, these non-song entries were manually identified through an examination of the song titles and subsequently removed.

Upon inspecting the downloaded song lyrics, we observed the presence of extraneous text in the beginning or end of the text related to metadata or the number of contributors, which was removed from the lyrics text. Additionally, junk characters and non-printing characters except for the newline character were removed to streamline the text. The resulting cleaned lyrics text was then utilized for the subsequent task of keyword generation.

4.2 Keywords

We sought to generate 3-5 non-synonymous keywords that represented the theme and meaning of

each song. We used the chat completion feature of the GPT-3.5-Turbo model to extract keywords from song lyrics. To generate the keywords, we performed prompt tuning using the ChatGPT API.

4.2.1 Prompt Engineering

We initially asked the model to generate the theme of the song, but it generated a summary instead of keywords. We then modified the prompt to include the term "keywords" and the number of keywords to generate.

To ensure that the model generated a different number of keywords each time, we modified the prompt to include phrases such as "less than 6 keywords" and "no more than 5 keywords." However, we found that the model consistently generated exactly 5 keywords. The wording that worked best for this was "using 3 to 5 keywords". We also found that some of the keywords generated were not single words but phrases. To address this, we explicitly mentioned "single word keywords" in our prompt.

Additionally, the model sometimes generated the word "keywords" in the output, so we added a requirement that the output must not contain this word. Even after including this information in the prompt, we found that the format of the output was not always consistent. The model sometimes generated the keywords as bullet points, sometimes separated by a comma, etc. So we modified the query to mention the format of the output- 'the keywords are separated by commas.' Despite including this information in the prompt, the outputs generated by the model were not uniform. To address this issue, we turned to few-shot prompting. We used the same prompt as before, but with 2 demonstrations. However, we noticed that the model sometimes summarized the songs in the demonstrations before generating the keywords.

To overcome this, we rephrased our input in a "Q&A" format as follows:

> Q: Describe the song with the following lyrics using 3 to 5 single word keywords, where the output is only the keywords separated by commas.
> ≪Lyrics of the song≫
>
> A: ≪Keywords≫

This format, combined with few-shot prompting, resulted in consistent and uniform generations by the model. We also set the system role content to reiterate all the information and provide clear instructions: *"You are a helpful assistant who is an expert in generating the theme of a song in 3 to 5 single word keywords from its lyrics. You must generate only the keywords."*

4.3 Dataset Examples

> **Lyrics:**
> *I walked through the door with you, the air was cold*
> *But somethin' 'bout it felt like home somehow*
> *And I left my scarf there at your sister's house*
> *And you've still got it in your drawer, even now*
> ....
>
> **Keywords:** memories, nostalgia, love, heartbreak, reflection

Example 1: Lyrics from the song "All Too Well (10 Minute Version) (Taylor's Version)"

> **Lyrics:**
> *I don't like your little games*
> *Don't like your tilted stage*
> *The role you made me play*
> *Of the fool, no, I don't like you*
> ....
>
> **Keywords:** revenge, betrayal, transformation, individualism

Example 2: Lyrics from the song "Look What You Made Me Do"

5 Baselines

We adopted an n-gram model with maximum likelihood estimation (MLE) as our baseline model. This baseline is suitable as lyrics often exhibit short-term dependencies and repetitive patterns within short phrases or lines. n-gram models excel at capturing these local dependencies by considering the preceding $n-1$ words to predict the next word. This allows them to generate coherent and grammatically correct short phrases or lines, which are essential in lyrical compositions.

To evaluate the performance of the n-gram model, we experimented with two versions of the model. First, we generated lyrics based on the first line of the lyrics, which is the traditional approach

for text generation. Second, we trained the model to generate text based on the corresponding keywords of the song passed as the first line, which is more aligned with our use case. We tested the two models for values of n ranging from 3 to 7, which was determined based on the average number of words in a single line of a lyric.

5.1 Baseline 1: Lyric generation given a first lyric line

To train the n-gram model, we utilized 80% of the dataset, which consisted of 262 songs. During the training process, we included the '\n' character as a token to facilitate the generation of structured lyrics. Subsequently, the trained model was employed to generate songs based on the first line of the lyric. The number of words to be generated was specified as a parameter for the model.

Initially, we attempted to incorporate an end of sequence token during the training phase. However, this approach resulted in a recurring issue where the end of sequence token was generated repeatedly after a few words were generated. To address this problem, we made the decision to exclude the end of sequence token from the training process. Consequently, we evaluated the model's performance by generating 100 words for any given prefix.

Input Line: "Walking downtown shuffling my feet"
Lyrics:
Walking downtown shuffling my feet
right down the aisle like a pageant queen
But I come back stronger than a trend
Wait for the signal and I meet you when you are worlds away
In New York no shoes
Looked up at the sky and it was summer and
I hate that stupid old pickup truck you never let me drive
You a bandit like me
And you said it reminded you of me
Baby that the fun of you Makin fun of you
Baby let the games begin
Are you ready for it

Example 3: Generated lyrics for n=4

Input Line: "The moon was as full as a hot cup of coffee"
Lyrics:
The moon was as full as a hot cup of coffee
tryna get to the phone right now
Was what I did to you
forgiveness is a nice thing to do
If there no you
Call a cab lose my number
You about to lose your face
look at your face Look at your face Look at your face gorgeous
And I said Oh my my my
Oh you took me with you
I never saw you comin
And I ca talk to you
Take me back to Highgate met all of his best mates
So I

Example 4: Generated lyrics for n=7

As we can see in Examples 3 and 4, the model exhibits the ability to generate individual lines that are coherent. However, when considering the lyrics as a whole, they lack overall coherence and fail to convey a meaningful message.

5.2 Baseline 2: Lyric generation based on keywords

To integrate keyword-based lyric generation into our approach, we introduced the keywords as the first line of each song. Subsequently, an n-gram model was trained using the modified dataset. This updated model demonstrated improved relevance between the generated lyrics and the provided keywords, as compared to the previous model.

Keywords: betrayal, disappointment, denial, regret
Lyrics:
If you were around
Too young to wander London streets alone and haunted
Born into nothing
With them I have something something to cling to
Visions of dazzling rooms I never get let into
And it moving so fast
Though I want to know how to cover up a scene
Good thing Este sister gon na swear she was with me
Never told me why

Never had to see me cry
Insincere apology so he does look like the bad ones too
I with you
I

Example 5: Generated lyrics for n=4

Keywords: bullying, vulnerability, empathy
Lyrics:
slamming door
And all the things that I misread
So babe if you know everything
Tell me why you could see
When I left I wanted you to chase after me yeah
I said Leave but all I really want is you
To stand outside my window throwing pebbles
Screaming I in love with you
Wait there in the pouring rain coming back for more
And do you leave I know all I need is on
The other side of the door
Me and my stupid pride

Example 6: Generated lyrics for n=7

The n-gram model trained on keyword data incorporates the mentioned themes from the input keywords, as demonstrated in Examples 5 and 6. This baseline model aligns with our specific application requirements. As a result, we conducted a comparative analysis between the performance of the n-gram model trained on keyword data and our proposed approach in the subsequent sections of this study.

Overall, increasing the value of n in the n-gram model results in improved coherence in the generated lines, as the model can capture more contextual information from the preceding $n-1$ words. Both versions of the n-gram model are limited to the vocabulary present in the training set, leading to the generation of existing lyrics.

6 Our Approach: Lyric Generation using T5

The Google Flan T5-Small model (Chung et al., 2022) was fine-tuned to generate the lyrics. It is an enhanced version of T5 that has been fine-tuned on a mixture of tasks. As we were unable to fine-tune the larger variations of T5 due to computational power restrictions, this model was suitable for this task.

The final training dataset consists of 319 songs, which were split into an $8:1:1$ ratio for training, validation, and testing.

To fine-tune the model, we created the input by modifying the training dataset to include the separator token '<sep>' between the keywords and the end of sequence token '</s>' at the end of the keywords and lyrics. The input for the model was the keywords, and the target was the lyrics. The maximum length of the input was set to 10 tokens (including <sep> and </s>), since the number of keywords ranged from $3-5$, and the maximum length of the target was set to 512 tokens.

We fine-tuned the model using a batch size of 16 for 120 epochs. The learning rate was changed every 40 epochs. The first 40 epochs had a learning rate of 0.001, the next 40 epochs had a learning rate of 0.0005, and the final 40 epochs had a learning rate of 0.0001. The learning rate was gradually reduced to ensure that the model does not overfit.

The training loss at the end of 120 epochs was 0.39577. Any further training, even with a much smaller learning rate, resulted in an increase in the training loss. Figure 1 shows the plot of the training loss against the number of steps.

7 Evaluation

7.1 Semantic Evaluation

In order to ensure that the generated lyrics were based on the input keywords, semantic evaluation was performed. The keywords extracted from the model generated lyrics were compared to the input keywords to determine their similarity.

The first step in this process was to extract non-synonymous keywords from the lyrics generated by the model using the same setup as described in 4.2. Both the input and generated keywords were then converted into embeddings using the pre-trained Google News 300 Word2Vec model (Mikolov et al., 2013a,b).

Next, the cosine similarity between each of the generated and input keywords was computed. For each generated keyword, the similarity score was assigned as follows:

Let i_1, i_2, i_3 represent the input keywords, and g_1, g_2, g_3 represent the keywords extracted from

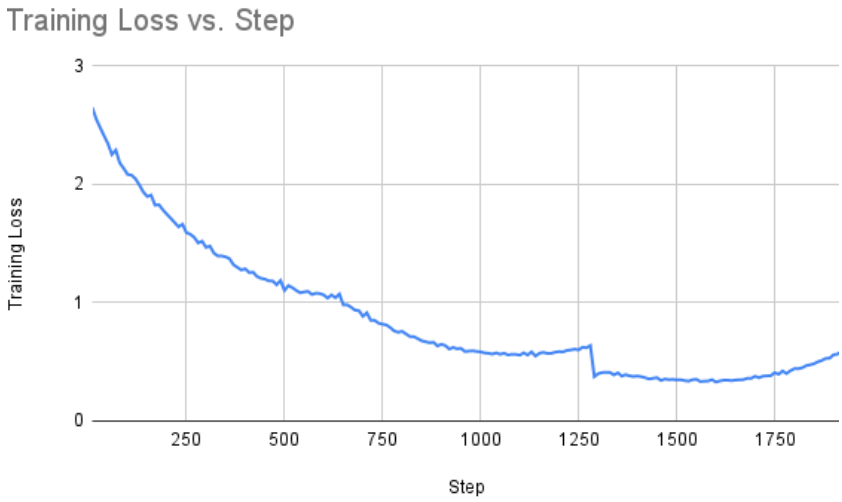

Figure 1: Plot of Training Loss vs Steps for FLAN-T5-small

the model generated lyrics.

$$score(g_j) = \max\Big(CosineSimilarity(g_j, i_1)\ CosineSimilarity(g_j, i_2),\ CosineSimilarity(g_j, i_3)\Big)$$

for $j = 1, 2, 3$

Once the maximum similarity score for each generated keyword was determined, the average, maximum and minimum of these scores were taken for each lyric. Finally, three semantic scores were computed for the test set- the mean of the average, maximum and minimum scores.

7.2 Stylistic Evaluation

To evaluate the similarity of the generated lyrics to Taylor Swift's style, we fine-tuned a BERT model as a binary classifier. The training data consisted of lyrics from 319 Taylor Swift songs and 300 songs from other artists across various genres like rap, country, pop, and rock. The dataset was evenly split to ensure balance during training.

The task was framed as a binary classification, determining whether a given set of lyrics belonged to Taylor Swift's style (class 1) or not (class 0). The BERT model was fine-tuned with specific hyperparameters, including a batch size of 16, a learning rate of 5×10^{-5}, an Adam epsilon of 10^{-8}, and 10 epochs. During training, the model's loss decreased, but the validation accuracy remained consistently at 1 from the first epoch, suggesting potential overfitting. To mitigate this, we reduced the learning rate from 5×10^{-3} to 5×10^{-5} and decreased the number of epochs from 35 to 10. The training loss has been plotted in Figure 2

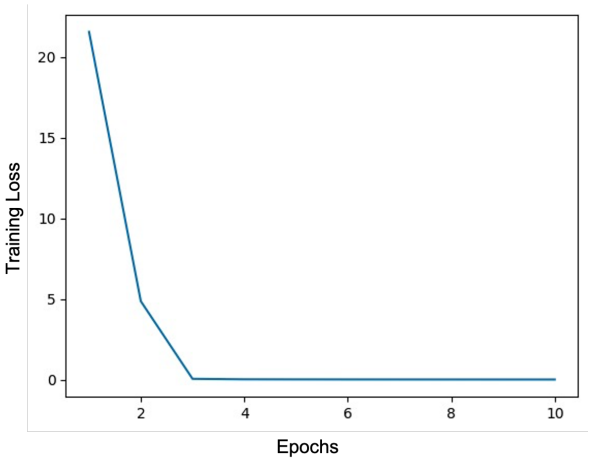

Figure 2: Plot of Training Loss vs Epochs for fine-tuned BERT for Stylistic Evaluation

8 Results and Analysis

8.1 Qualitative Analysis

Qualitative analysis was performed on the lyrics generated by our proposed model and the baseline models to compare their coherence, alignment to the input keywords and overall standard. We found that the fine-tuned Flan-T5-Small model produced lyrics of a higher quality.

The primary limitation of the n-gram model is its reliance on predicting the next word solely based on its vocabulary, which is wholly derived from the training dataset. Consequently, the generated lyrics are often identical to existing ones and lack relevance to the keywords. In contrast, the

Flan-T5-Small model possesses a much broader knowledge base, enabling it to generate unique lyrics that are more contextually connected to the provided keywords.

To support this finding, we present two examples featuring keywords and the corresponding lyrics generated by the n-gram model with $n = 7$ and the fine-tuned Flan-T5-Small model (only the unique lines generated have been included for the Flan-T5-Small model outputs).

Keywords: war, separation, family, longing

n-gram generated Lyrics:
it killed you just the same
Cursing my name wishing I stayed
Look at how my tears ricoche

I feel
The lavender haze creeping up on me
Surreal
I damned if I do give a damn what people say
No deal No deal
The shit they want from me
I just wan na stay in that lavender haze

That lavender haze

Talk your talk and go viral
I just need this love spiral
Get it off your chest
Get it off my desk

Fine-tuned FLAN T5-small generated Lyrics:
The battle was going to go down The battle was going to take place And the next day, the two of us were gone And the next day, we were gone And the baby came out crying And crying all night long And the battle was going to go down Then back to the family We were leaving and leaving in a war Then the family came and told him to stay And he said, "It's not enough" And he said, "It's not enough"

Example 7

The n-gram generated lyrics in Example 7 repeat existing lyrics directly, such as the phrases *'I feel The lavender haze creeping up on me'* and *'Talk your talk and go viral'* from the song 'Lavender Haze', whereas the Flan-T5-Small model is able to generate unique and new lyrics.

The lyrics generated by the n-gram model only represent separation and longing through the following line - *'it killed you just the same Cursing my name wishing I stayed'*. However, the fine-tuned model is able to capture all of the keywords - *'the battle'* is similar to the war theme, there is a mention of family, and separation and longing are indicated by the want to leave and the line *'It's not enough'*.

Keywords: nostalgia, memories, bonding

n-gram generated Lyrics:
was nothing left to do
When the butterflies turned to
Dust that covered my whole room
So I punched a hole in the roof
Let the flood carry away all my pictures of you

The water filled my lungs I screamed so loud
But no one heard a thing
Rain came pouring down
When I was drowning that when I could finally breathe
And by morning
Gone was any trace of you I think I am finally clean

Oh oh
Finally clean
Think I finally

Fine-tuned FLAN T5-small generated Lyrics:
I remember you driving down the river side with your little brother And his big city truck on the ground You hit the ground running and running and running and running And you were sitting there in front of me You put your arm around me, holding my hand I remember you driving down the river side You took me by the hand and then you took me down That's a shame, I'm not the one I've never seen before And I remember how you watched me walk down the river You took me by the hand and then you took me down That's a shame, I'm not the one I've never seen before

Example 8

The n-gram model generated lyrics are almost entirely from the song 'Clean', whereas the Flan-T5-Small model is able to generate novel lyrics. The n-gram model only captures the keyword 'memories' through the phrase *'pictures of you'*.

Meanwhile, the fine-tuned Flan-T5-Small model is able to represent all of the keywords. The line *'You put your arm around me, holding my hand'* captures the bonding, the reference to the drive down the riverside represents the memories, and the phrase *'That's a shame, I'm not the one'* mentioned twice indicates longing and nostalgia. The recurrence of the phrase *'I remember'* suggests the presence of an anaphora, a literary device frequently employed in Taylor Swift's songs. This observation implies that the Flan-T5-Small model may have successfully captured and incorporated such lyrical characteristics into its generated lyrics.

8.2 Quantitative Analysis

8.2.1 Semantic Analysis

The test set consisted of 10% of the dataset- 33 keywords and song pairs. The maximum, minimum and average similarity scores were computed between the keywords extracted from the generated lyrics and the input keywords for each song in the test set. Table 1 shows the comparison of the mean minimum, maximum and average similarity scores of our approach and the baseline n-gram model trained with keywords (5.2) on the test set for semantic evaluation, as discussed in 7.1.

The fine-tuned Flan-T5-Small model performs better than the baseline n-gram models in all three similarity scores (except for the mean of the maximum scores for $n = 3$ n-gram). The keywords extracted from the generated lyrics are non-synonymous, thus the mean of minimum scores is the more important metric for our problem statement because the generated lyrics must be representative of all the keywords, not just a subset. The proposed model outperforms the $n = 7$ n-gram model, which generated the most coherent lyrics of all the baseline models, by 0.090372 in this metric.

Based on these results, we can conclude that the lyrics generated by the Flan-T5-Small model is able to better capture the keywords compared to the n-gram model, and thus produces lyrics of higher quality.

8.2.2 Stylistic Analysis

We evaluated the lyrics generated by both the n-gram and Flan-T5-Small models. The generated lyrics underwent classification by the fine-tuned BERT classifier. However, the classifier consistently predicted class 0 for all generated lyrics, indicating a deviation from Taylor Swift's style.

This outcome can be attributed to two possible factors. Firstly, despite adjustments made during training, the model may still be overfitting to the data. However, the model achieved an accuracy of 1.0 on both the validation and test sets, disproving this possibility. The other and more probable reason is that the quality of the generated lyrics from our tested models may have been subpar, leading to the classifier predominantly predicting class 0. Therefore, it is crucial to explore further enhancements to the quality of the generated outputs in order to validate the effectiveness of this classifier in identifying Taylor Swift's style.

9 Error analysis

The poor outputs or errors generated by the proposed model and baseline models can be categorised into the following:

1. Lyrics on keywords representing themes outside the training dataset
2. Lyrics on a set of dissimilar keywords
3. Line repetition and song structure in lyrics

9.1 Lyrics on keywords representing themes outside the training dataset

The n-gram model struggles to generalize to new themes due to its vocabulary being solely derived from the training dataset, as mentioned earlier. Whereas, the Flan-T5-Small model exhibits good generalization to keywords representing themes that were not present in its training dataset. This suggests that the model can effectively generate lyrics for a broader range of themes beyond its training data. Examples 9 and 10 support this finding.

Keywords: earth, greenery, nature, peace

n-gram generated Lyrics:
Would it be enough if I could never give you peace
Your integrity makes me seem small
You paint dreamscapes on the wall

	Model	Maximum	Minimum	Average
Our approach	Fine-tuned FLAN-T5-small	0.665183	0.247981	0.426343
Baseline	n-gram: $n = 3$	0.67327	0.16441	0.357993
	n-gram: $n = 4$	0.61822	0.158776	0.340113
	n-gram: $n = 5$	0.604615	0.153883	0.333427
	n-gram: $n = 6$	0.641971	0.176509	0.352843
	n-gram: $n = 7$	0.590014	0.157609	0.314976

Table 1: Comparison of similarity scores of baseline model and our approach

I talk shit with my friends it like I wasting your honor
And you know that I swing with you for the fences
Sit with you in the trenches
Give you my wild give you a child
Give you the silence that only comes when two people understand each other
Family that I chose now that I see your brother as my brother
Is it enough

Keywords extracted from the generated lyrics: insecurity, admiration, commitment, family, doubt

Example 9: Baseline model - $n = 7$ n-gram model

Semantic Scores:

- **Maximum Similarity =** 0.2945196
- **Minimum Similarity =** 0.11715406
- **Average Similarity =** 0.19994505

Keywords: earth, greenery, nature, peace

Fine-tuned FLAN T5-small generated Lyrics:
Greenery is the fun of the seasons Playing with my guitar and chasing the ball And sometimes, I don't like the way it's tossed in clouds But it's nice to see the stars running wild It's like the greenery on the ground It's like the breeze on the ground It's like it's getting colder on me It's nice to see the sun running wild

Keywords extracted from the generated lyrics: nature, joy, freedom

Example 10: FLAN-T5-Small model

Semantic Scores:

- **Maximum Similarity =** 1.0
- **Minimum Similarity =** 0.2258664
- **Average Similarity =** 0.53679657

The lyrics generated by the proposed model achieves a higher score for all three similarity metrics. The n-gram model is only able to capture the keyword 'peace' in its lyrics. Whereas, the Flan-T5-Small model is able to clearly represent the keywords 'greenery' and 'nature', and by extension 'earth.' It is also able to represent the keyword 'peace' subtly through the activities mentioned- *'Playing with my guitar'* and *'chasing the ball'*.

9.2 Lyrics on a set of dissimilar keywords

Both the proposed model and the n-gram models struggle to generate lyrics for keywords that are distinct from each other and do not indicate a single theme. This finding is supported by Examples 11 and 12.

Keywords: matrices, car, tennis

n-gram generated Lyrics:
With some thing
And you asked me to dance
But I said Dancin is a dangerous game
Oh I thought
This is gon na be one of those things
Now I know
I never gon na love agai
Let you know that what I feel is true
And I only me
Who I wan na be
Well I only me when I with you
When I with anybody else
It so hard to be myself
And only you can tell

Keywords extracted from the generated lyrics: court, dance, danger, love, authenticity

Example 11: Baseline model - $n = 7$ n-gram model

Semantic Scores:

- **Maximum Similarity =** 0.2587057
- **Minimum Similarity =** 0.051453896
- **Average Similarity =** 0.1552248

Keywords: matrices, car, tennis

Fine-tuned FLAN T5-small generated Lyrics:
You walk in the car with your arms crossed You're on the floor with a chalkboard And you're on the floor with a big screen TV You watch the game with your friends And you watch the game with your friends

Keywords extracted from the generated lyrics: boredom, repetition, routine

Example 12: FLAN-T5-Small model

Semantic Scores:

- **Maximum Similarity =** 0.11720182
- **Minimum Similarity =** 0.11362423
- **Average Similarity =** 0.11515386

The n-gram model exhibits higher maximum and average similarity scores when compared to the proposed model. Conversely, the proposed model demonstrates a higher minimum similarity score. This provides additional evidence that the Flan-T5-Small model effectively captures more keywords in its generated lyrics.

The lyrics generated by the Flan-T5-Small model explicitly mention the term 'car', while the keyword 'matrices' is very subtly referenced through the mention of *'chalkboard'* in the lyrics. Similarly, the keyword 'tennis' is very subtly indicated through the phrase *'game on the TV.'*

On the other hand, the lyrics generated by the baseline model only partially capture the keyword 'tennis', as indicated by the word *'game.'* However, in the context of the lyrics, the word *'game'* does not refer to the sport itself but rather to dancing. Consequently, the baseline model fails to capture any of the keywords effectively.

This flaw in the proposed model is not due to the lack of such examples in the training dataset. One such example is the song 'Cruel Summer' from which the following keywords were generated: summer, desire, love, pain, secrecy. However, when presented with distinct keywords 'blame, hurt, trust, possibilities' during testing, the generated lyrics achieved the lowest semantic scores in the test set.

9.3 Line repetition and song structure in lyrics

The proposed model is only able to generate at most 7-8 unique, coherent lines in the lyrics before repeating the last line until the maximum target length is reached.

The Flan-T5-Small model was unable to generate the newline character to separate the lyrics into the traditional line-verse structure, while the n-gram model was able to generate separate lines of lyrics.

We attribute these flaws to the small size of the model. Due to the availability of minimum computational resources, we could only train the small version of the Flan-T5 model, rather than the T5-base model.

10 Contributions of group members

- Aditi: Worked on keyword extraction from lyrics, fine-tuning T5 model, semantic evaluation
- Niel: Worked on lyrics extraction, training BERT classifier for stylistic evaluation, error analysis
- Suraj: Worked on building the baselines, semantic evaluation of baselines, error analysis
- Akshaya: Worked on lyrics extraction, data pre-processing, analysis and evaluation

11 Conclusion

In this study, our aim was to generate song lyrics in the style of Taylor Swift based on input keywords and themes. We utilized a fine-tuned Flan-T5-Small model, trained on Taylor Swift song lyrics and their corresponding keywords extracted using

GPT 3.5, as our primary approach. As a baseline, we employed n-gram models for comparison. To assess the quality of the generated lyrics, we conducted evaluations focusing on both stylistic and semantic aspects.

For stylistic evaluation, we attempted to train a BERT model to classify whether a given set of song lyrics was written by Taylor Swift. However, we encountered challenges as the BERT model was ineffective in classifying our generated lyrics. On the other hand, for semantic evaluation, we measured the alignment between the input keywords and the keywords extracted from the generated lyrics, using a similarity score. Our findings indicated that the Flan-T5-Small model outperformed the n-gram baselines in terms of generating more coherent and unique lines that were relevant to the input keywords.

Based on our qualitative analysis, the Flan-T5-Small model demonstrated its capability to generate lyrics that exhibited higher coherence and uniqueness, aligning well with the input keywords. Nonetheless, we observed a tendency for repetitive content beyond the initial lines in the output of our proposed model. These outcomes are promising, and future work should explore avenues for improving stylistic evaluation, including the development of a more robust classifier to identify lyrics in the style of a specific artist.

Moving forward, we also plan to address the issue of repetitiveness by employing larger computational resources and training a larger language model such as T5-base. It is expected that with increased computational capabilities, we will be able to enhance the quality of the generated lyrics and achieve more desirable outcomes in terms of creativity, uniqueness and improved stylistic similarity.

Our models, code and data files can be found in the following Github Link: Keyword Based Lyric Generation

12 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes, ChatGPT

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - What are some general guidelines for the problem statement section of a paper/project report?

    Help me rephrase my work for the problem statement section based on the guidelines you mentioned (and other such prompts to rephrase content for an academic report)
  - Why n-gram is a good baseline for lyric generation
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - It was helpful. It was only used to rephrase text to suit an academic report.

References

Briakou, E., Agrawal, S., Tetreault, J., and Carpuat, M. (2021). Evaluating the evaluation metrics for style transfer: A case study in multilingual formality transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1321–1336, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling instruction-finetuned language models.

Gill, H., Lee, D., and Marwell, N. (2020). Deep learning in musical lyric generation: An lstm-based approach.

Malmi, E., Takala, P., Toivonen, H., Raiko, T., and Gionis, A. (2015). Dopelearning: A computational approach to rap lyrics generation. *CoRR*, abs/1505.04771.

Manjavacas, E., Kestemont, M., and Karsdorp, F. (2019). Generation of hip-hop lyrics with hierarchical modeling and conditional templates. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 301–310, Tokyo, Japan. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.

Miller, J. W. (2017).

Miller, J. W. (2020).

Nikolov, N. I., Malmi, E., Northcutt, C. G., and Parisi, L. (2020). Rapformer:conditional rap lyrics generation with denoising autoencoders. *CoRR*, abs/2004.03965.

Nikzad-Khasmakhi, N., Feizi-Derakhshi, M., Asgari-Chenaghlu, M., Balafar, M. A., Feizi-Derakhshi, A., Rahkar-Farshi, T., Ramezani, M., Jahanbakhsh-Nagadeh, Z., Zafarani-Moattar, E., and Ranjbar-Khadivi, M. (2021). Phraseformer: Multimodal key-phrase extraction using transformer and graph embedding. *CoRR*, abs/2106.04939.

Potash, P., Romanov, A., and Rumshisky, A. (2015). 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. Association for Computational Linguistics.

Qian, Y., Jia, C., and Liu, Y. (2021). Bert-based text keyword extraction. In *Journal of Physics: Conference Series*, volume 1992, page 042077. IOP Publishing.

Ram, N., Gummadi, T., Bhethanabotla, R., Savery, R. J., and Weinberg, G. (2021). Say what? collaborative pop lyric generation using multitask transfer learning. *CoRR*, abs/2111.07592.

Vechtomova, O., Bahuleyan, H., Ghabussi, A., and John, V. (2018). Generating lyrics with variational autoencoder and multi-modal artist embeddings.

Watanabe, K., Matsubayashi, Y., Fukayama, S., Goto, M., Inui, K., and Nakano, T. (2018). 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. Association for Computational Linguistics.

Xue, L., Song, K., Wu, D., Tan, X., Zhang, N. L., Qin, T., Zhang, W., and Liu, T. (2021). Deeprapper: Neural rap generation with rhyme and rhythm modeling. *CoRR*, abs/2107.01875.