Language Modeling to Generate Lyrics for Hip-Hop and Gospel Songs

Bryon Kucharski bkucharski@umass.edu
Damain Moquin dmoquin@umass.edu
Juexing Wang juexingwang@umass.edu
Weiqiu You wyou@umass.edu

1 Problem statement [1]

The main goal of our project is to generate song lyrics for the Hip-hop and Gospel genres. For a given genre and start word, we generate a full song with multiple verses in that genre. We want our lyrics to be as coherent and meaningful as possible, ideally indistinguishable from human-written lyrics in these aspects. To do this, we implemented two RNN models trained on Gospel and Hip-Hop song lyrics respectively.

We think that lyric generation can be very useful for musical artists that have writers block. If a songwriter has no idea what to write about or is unsure on how they want to word a line or verse, they can generate new songs to spark their creativity and give them new ideas. Lyric generation is still very difficult for even the most advanced natural language models to get right. There are many aspects of songwriting to consider such as rhyme scheme, syllable count, melody, rhythm, theme, wordplay, etc. For this reason, we are not attempting to replace the musical artist with our model; we want to assist them in the songwriting process by giving them new ideas. For now at least, there is still an aspect of human creativity and musical knowledge that cannot be emulated artificially.

The Hip-Hop/Gospel split has been a very interesting aspect of our project. The varied musical interests within our group was a major factor in this split. There are also many large differences between the Hip-Hop and Gospel genres. They differ in lyrical content, choice of themes, repetition, lyrical complexity, target audience, and much more. As a side goal, we wanted to figure out how these differences would affect song generation, and whether one genre was more complex or difficult to model than the other. To do this, we preformed some data analysis on our dataset and representative songs of the two genres, and we carefully looked for differences in the output of our models for each genre.

2 What you proposed vs. what you accomplished

- ~~Create hip hop dataset~~
- ~~Create gospel dataset~~
- ~~Create bi-gram model~~
- ~~Create tri-gram model~~
- ~~Create LSTM model~~
- Give input sentences to each model and observe output

As explained in later sections, we found that it may not be fair to ask a model to generate something that it is not trained on, meaning training on gospel data and asking to generate lyrics not related to God. Thus, we decided to not pursue the original plan to give each model the same start words.

3 Related work

Nguyen and Sa (2009) created a rap lyrics generator that makes new rap lines based on an existing corpus of rap songs, and emulates the rhyme and syllable structure of the genre. Their dataset was a database of over 40,000 rap song lyrics scraped from a hip-hop lyrics website. They sampled from an interpolated quadragram language model to generate their lyrics. Much of our approach was inspired by this method, from the lyric web-scraping dataset collection to the n-gram model for lyric generation. Their discussion about attempting to create specific themes for lines and verses inspired us to attempt using start words to

[1] All code and example output for each model can be found at https://github.com/bryonkucharski/Language-Modeling-to-Generate-Lyrics-for-Hip-Hop-and-Gospel-Songs

make lyrics conform to a theme. However, our approach differs from theirs in significant ways and tries to expand upon their method. We generated Gospel as well as Hip-Hop/Rap lyrics so that we could examine how our results differed for each genre, and so that we could determine if one genre might be harder to model lyrics for than another. We also used n-gram models as our baseline, but we decided to implement a LSTM model to improve the coherency and meaning of the lyrics. We opted not to use a rhyming database or directly enforce a rhyme scheme on the lyrics, to keep our model relatively simple but also because enforcing rhymes could negatively impact aspects like coherency and meaning.

Watanabe et al. (2018) proposed a model for generating lyrics based on a given input melody. They utilized an RNN language model trained on over 54000 Japanese song lyrics as well as 1000 lyric/melody pairs created using digital music scores available online. For the song lyrics, they created pseudo-melodies by using the distribution of notes, rests, and pitch sampled from the lyric/melody pair data. Both this project and ours sought to generate song lyrics for musical artists, but theirs was much more focused on the relationship between lyrics and melody while ours focused on the differences in genres. Both approaches used perplexity and human evaluation as metricc of evaluating the output of our models. Using melody as a means of data for the model is interesting and does seem to increase the listenability and flow of the lyrics, but their results also suggest that it might slightly negatively impact the coherence and meaning of their lyrics. We also believe there is value in the ability to generate a song without an input melody in mind.

There are multiple other papers that describe approaches for generating lyrics based on melody. Oliveira et al. (2007) created a model that generates lyrics for an existing melody, using their analysis on the relationships between lyrics, melody, beats, tempo, and rhyme to select vocabulary. Nichols et al. (2009) used melody-lyrics parallel data to investigate the correlation between things like syllable stress and pitch. Ramakrishnan et al. (2009) proposed a statistical model to generate Tamil lyrics for a melody, utilizing Dijkstra's Shortest Path Algorithm and melody-lyrics data.

There are other similar projects that focus on different aspects of lyric generation and contain ideas potentially worth implementing in our project in the future. Barbieri et al. (2012), Abe and Ito (2012), and Davismoon and Eccles (2010) all proposed models for generating lyrics under various constraints like rhythm, rhyme, and part of speech. Hirjee and Brown (2010) developed a rhyme detection tool using a probabilistic model, and analyzed phonetic patterns in words. Potash et al. (2015) used an LSTM as well as an n-gram model to generate lyrics in the style of a specific Hip-Hop artist. Malmi et al. (2015) generated fixed 16-line rap verses line by line by sampling full lines from existing rap songs. There are many different approaches to lyric generation as well as relationships to analyze in lyric data.

4 Our dataset

4.1 Data Collection

To begin the data collection, we first needed a list of artists to collect lyrics for. To do this, we gathered all of the names from the List of Hip Hop Artists Wikipedia list for the Hip-Hop artists and all names from the List of Christian Worship Music Artists Wikipedia list for Gospel artists into a text file. With this text file, we used a python library called PyLyrics3 to scrape LyricWiki website. This library saves lyrics for each artist's songs as text files in a subdirectory for that artist.

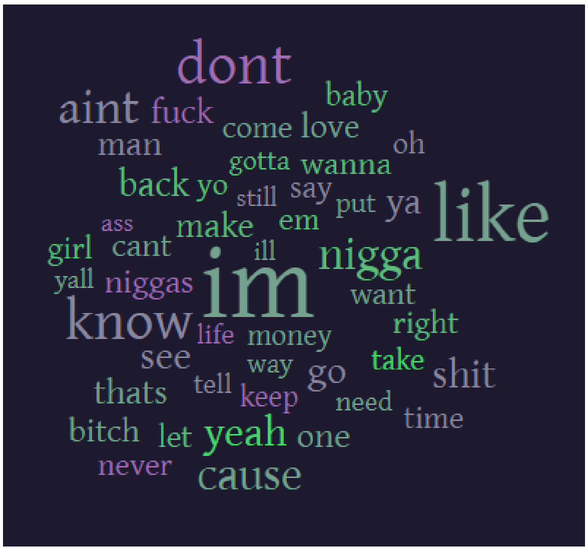


Figure 1: The top 50 most common words in our hip-hop corpus, excluding some common stop words like "the", in word cloud format. Size is proportional to the words relative occurrence rate. The most common word was "I'm", with 168,059 appearances, and the fiftieth most common word was "ass", with 18,601 occurrences.

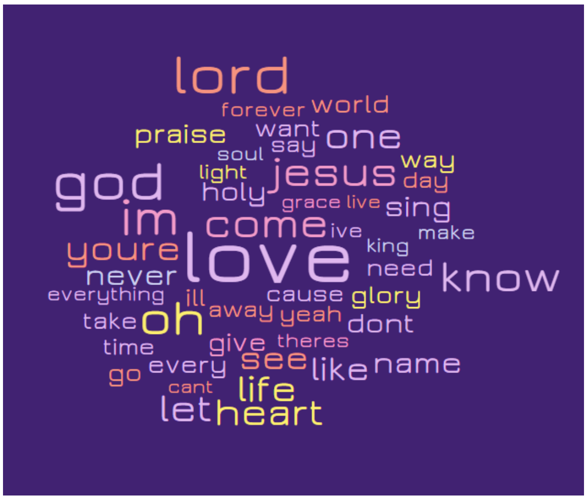


Figure 2: The top 50 most common words in our gospel corpus. The most common word was "love", with 18,832 occurences, and the fiftieth most common word was "live", with 2,997 occurrences.

4.2 Data Prerocessing

Once we collected all lyrics, we needed to clean the data to use with our algorithms. Each song was preprocessed by:

1. Removing the punctuation from each token
2. Removing tokens that are not alphabetic
3. Make all tokens lower case

These steps ensure that the same words with different punctuation, annotation, or capitalization are treated as the same one word. For example, "that", "That", "that,", and "that-", all become "that" after preprocessing. This ensures that things like word counts are accurate for our model to train on.

Then, a training set and test set was created by designating 20 percent of the songs for testing and 80 percent of the songs for training randomly in each genre's dataset. The training set was for our model to train on, and the test set was used to test our model's performance. We made sure not to train our models on the test set.

4.3 Data Analysis

Our dataset consists of 28063 hip-hop and 9945 gospel songs. Below are some example songs from both genres. They are contained in our dataset as well as the billboard top 50 for their genre from 2017. We believe that they are fairly representative of the major tropes of each genre.

Excerpt from "Bank Account"

> I pull up in 'Rari's and shit, with choppers and Harley's and shit (for real)
> I be Gucci'd down, you wearing Lacoste and shit (bitch)
> Yeah, Moncler, yeah, fur came off of that, yeah (yeah)
> Triple homicide, put me in a chair, yeah (in jail)
> Triple cross the plug, we do not play fair, yeah (oh God)
> Got 'em tennis chains on and they real blingy (blingy)
> Draco make you do the chicken head like Chingy (Chingy)
> Walk in Neiman Marcus and I spend a light fifty (fifty)
> Please proceed with caution, shooters, they be right with me (21)
> Bad bitch, cute face and some nice titties

Figure 3: Excerpt from the first verse of "Bank Account" by 21 Savage, 2017 Hip-Hop Billboard number 23 and part of our dataset. The central topic is about the artist's wealth, but it also has lines about violence/murder ("triple homocide"), drug dealing ("triple cross the plug"), and women ("bad bitch, cute face"). The verses are lengthy, and there are many similes and references in the song.

Songs are separated into lines, which are part of verses. Typically there is a chorus or hook in most songs. There is also typically a rhyme scheme and a rhythm that is obeyed throughout the lyrics. A song is usually about a specific set of topics or themes, that differ based on the genre. In general, songwriting is a complex creative process that is difficult for even the most skilled musical artists. For this reason, generating coherent ad meaningful song lyrics is a daunting task.

Hip-hop and Gospel songs differ in many ways (e.g. lyrical content, choice of themes, repetition, lyrical complexity, target audience, etc). There are also some qualitative statistical differences between the two genres that we examined, and are demonstrated in the table below.

Since there are more popular Hip-Hop songs and artists than Gospel songs and artists, we have more data for Hip-Hop lyrics. Our Hip-Hop models thus had more data to train on than the gospel models. Hip-Hop has noticeably more lines per

Excerpt from ”Change Me”

Change me, oh God
Make me more like You
Change me, oh God
Wash me through and through
Create in me a clean heart
So that I may worship You

Change me, oh God
Make me more like You
Change me, oh God
Wash me through and through
Just create in me a clean heart
So that I may worship You
I need you to...

Figure 4: The first two verses of ”Change Me” by Tamela Mann, 2017 Gospel Billboard number 2. The song is about God purifying the artist, and it does not stray from this topic at all. The lines and verses are fairly short, and there is significant repetition of words and lines.

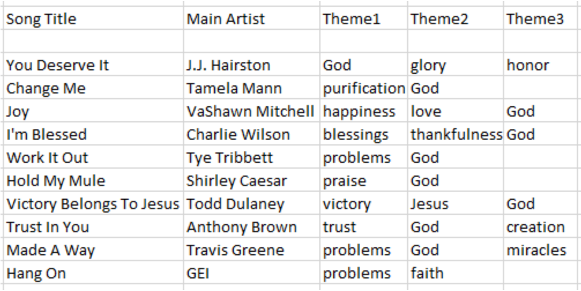
Genre dataset:	Hip-Hop	Gospel
Total number of Artists:	531	196
Total number of songs:	28063	9945
Total number of lines:	1831744	329348
Total number of words:	14335128	2003785
Average lines/song:	65.27	33.12
Average words/song:	510.82	201.49
Average words/line:	7.83	6.08
Avg. unique lines/song:	53.95	22.32
Unique lines/song (%):	82.65	67.39
Avg. unique words/song:	218.85	74.93
Unique words/song (%):	42.84	37.19

Figure 5: A table displaying some comparative statistics of our Hip-Hop and Gospel datasets. We have significantly more data for the Hip-Hop dataset (artists, songs, lines, words), Hip-Hop songs have a higher lyrical density than Gospel (lines/song, words/song, words/line), and Hip-Hop on average uses more unique lines and words than in Gospel.

song, as well as more words per line and per song on average than Gospel. Based on the percentages of unique lines and words, Hip-Hop also repeats less words and lines than Gospel on average as well. From this data, we prepose that Hip-Hop songs are more compositionally complex than Gospel songs on average. Overall, there is more data and more unique data for any language model we use on the Hip-Hop dataset then on the Gospel dataset.

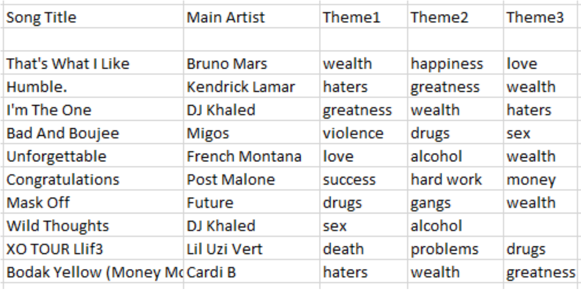
Our team also examined the lyrics of the Billboard Top 50 songs in Hip-Hop and Gospel for the year 2017. We discussed and recorded some of the top themes for each song, based on the lyrical content. Themes like ”God”, ”Jesus”, and ”glory” dominated the Gospel charts (with God being a theme in almost every top 50 song), while themes like ”money”, ”drugs/alcohol”, ”sex”, and ”violence” appeared frequently throughout the Hip-Hop top 50. Based on the diversity of themes we observed and recorded, it seemed to us that Hip-Hop songs varied in themes a lot more than Gospel songs, between and within songs. In fact, from our observation, Gospel songs tend to remain focused on one main topic for the entire song (usually God related), while Hip-Hop songs change topic repeatedly throughout the song, often from line to line. This actually makes it easier to generate coherent lyrics for Gospel, since most lyrics will tend to be God-related and relate to each other with ease, while in Hip-Hop it is difficult to jump from one topic to another while still keeping the coherence and central theme of the song in tact.

Another interesting observation from the Billboard Top 50 was that there are some shared themes between top songs of both genres, however the context for said themes differs significantly between genres. For example, the line ”I’ve got true love instead of pain” from VaShawn Mitchell’s ”Joy” is talking about true love for God, so a central theme for the song we decided on is ”love”. The lines ”And you are unforgettable / I need get you alone” from French Montana Ft. Swae Lee’s ”Unforgettable” refer to feelings of love/lust for a woman in a club, so a theme we decided on is ”love”. Both songs share the same theme, but the context of the theme differs significantly because of the difference in topics explored by both genres. Songs in Gospel that have ”greatness” as a theme usually talk about the greatness of God, while songs in Hip-Hop with that theme usually talk about the greatness of the artist’s life. Similarly, Gospel songs talk about God solving all of their ”problems”, while Hip-Hop songs talk about the ”problems” associated with the ghetto or drug trafficking.



Song Title	Main Artist	Theme1	Theme2	Theme3
You Deserve It	J.J. Hairston	God	glory	honor
Change Me	Tamela Mann	purification	God	
Joy	VaShawn Mitchell	happiness	love	God
I'm Blessed	Charlie Wilson	blessings	thankfulness	God
Work It Out	Tye Tribbett	problems	God	
Hold My Mule	Shirley Caesar	praise	God	
Victory Belongs To Jesus	Todd Dulaney	victory	Jesus	God
Trust In You	Anthony Brown	trust	God	creation
Made A Way	Travis Greene	problems	God	miracles
Hang On	GEI	problems	faith	

Figure 6: Top three themes for the top ten songs in the Gospel Billboard Top 50. God is a theme in almost all Gospel songs. Other themes like victory, problems, and faith are prominent in the Top 50 and the genre as a whole.



Song Title	Main Artist	Theme1	Theme2	Theme3
That's What I Like	Bruno Mars	wealth	happiness	love
Humble.	Kendrick Lamar	haters	greatness	wealth
I'm The One	DJ Khaled	greatness	wealth	haters
Bad And Boujee	Migos	violence	drugs	sex
Unforgettable	French Montana	love	alcohol	wealth
Congratulations	Post Malone	success	hard work	money
Mask Off	Future	drugs	gangs	wealth
Wild Thoughts	DJ Khaled	sex	alcohol	
XO TOUR Llif3	Lil Uzi Vert	death	problems	drugs
Bodak Yellow (Money Mc	Cardi B	haters	wealth	greatness

Figure 7: Top three themes for the top ten songs in the Hip-Hop Billboard Top 50. The themes tend to be more varied than in Gospel songs, but topics like wealth, drugs, and sex appear quite often.

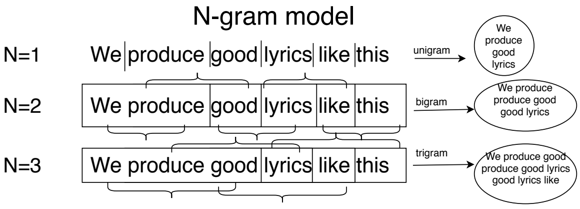

N-gram model
N=1
We produce good lyrics like this
unigram
We produce good lyrics
N=2
We produce good lyrics like this
bigram
We produce produce good good lyrics
N=3
We produce good lyrics like this
trigram
We produce good produce good lyrics good lyrics like

Figure 8: Example of N-gram models

5 Baselines

We implemented an N-gram model as our baseline algorithm. N-Gram is an algorithm which is based on a Statistical Language Model. Comparison between it and a Recurrent Neural Network Language Model(RNN/LSTM) could be meaningful. N-gram models focus on the relationship between a current word and the previous N-1 words and RNN/LSTM focus on the relationship between entire sentences and verses. We were wondering which model could generate more 'unartificial' verses. Since N-gram models are Statistical Language Models, we didn't apply train/validation/test splits to these models.

Our model can read in all the data files in ".txt" format first and preprocess this data to a dictionary type. After preprocessing, we will get a ".txt" file which contains the aggregated lyric data in a dictionary format. Our algorithm will detect whether this aggregate ".txt" file exists every time when run. Hence we can save a lot of time on processing data and get several results with one dataset, which will be helpful in our analyzing parts of our result.

Now, our N-gram model can generate songs in both Gospel and Hip-Hop tracks using either Bigram or Tri-gram two models. Our model has a parameter "lpv"(length per verse). It is correlated with length of verse in our dataset. We are trying to use some random or artificial "lpv" to replace the original "lpv" to control the length of verse which will be generated from model so that we can get random length songs as our wish. The only limitation in this part is the size of dataset; Hip-Hop works well but Gospel songs need more data to generate songs in a natural way. Another parameter "tpl" controls tokens per line. To make the sentence of our generated song look more unartificial, we won't apply random "tpl" data in this parameter.

6 Your approach

6.1 N-Gram

We implemented our N-gram model in Bi-gram and Tri-gram two ways. Bi-gram model

$$P(w_1, w_2....., w_m) = \prod_{i=1}^{m} P(w_i|w_{i-1})$$

Tri-gram model

$$P(w_1, w_2....., w_m) = \prod_{i=1}^{m} P(w_i|w_{i-2}w_{i-1})$$

Probability of Bi-gram model

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})}$$

Probability of Tri-gram model

$$P(w_i|w_{i-1}) = \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})}$$

We can treat song's word as key of dictionary and count total amount of each word to save it as the value of key. Then, we can compute the probability of each combination by using dictionary's value and then produce our output. We will treat each verse of output as a song since our preprocessed data didn't divide each song to several verse. So the 'lpv' will represent lines per song rather than lines per verse.

6.1.1 Bi-Gram Output

The most common length of Hip-hop songs in our data is 56 lines, 543 of total songs have 56 lines in their songs. Below is an example output of the hip hop Bi-Gram model

> Street smokin the horns off that you begin
> Look but that he comes
> Gotta some grown
> Stackin paper like plutonium in my homie
> His own
> You gotta hold your mind and when we still gonna come off my face
> The pimps man
> I feel
> Gone its the penn for the metal fist fulla keys

The most common length of Gospel in our data is 24 lines, 332 of total songs have 24 lines in their songs. Below is an example output of the gospel Bi-Gram model

> Somebody gets sweet sound here goes wild night i sing because
> Through me in waves and pain
> Yes i will rise and works
> At his heart is no limits
> No matter where would break this hill
> Sometimes hard to me
> Kyrie eleison down low
> And righteousness they shall come out the saints that i love
> You alone
> And worship you

6.1.2 Bi-Gram Discussion

The Bi-gram model in Hip-hop seems to work better than in Gospel. The generated songs of Hip-hop produce more meaningful sentences in a song since short sentences are more commonly produced in Hip-hop than in the Gospel model. Syntactic structures are bad in both types of output. The Bi-gram model only focuses on the current word and former word so it will ignore sentence structure and choose the combination with highest probability as output rather than considering syntactic structure and give some meaningless results with high probability.

6.1.3 Tri-Gram Output

Since we are using the same dataset on Bi-gram and Trigram, the number of lines is the same as above. Below is an example hip hop song using the Tri-Gram model

> im the best lucky bomb im finally so atat
> catch your body with some panties to the air
> well watch some head on your ass upstairs sometimes
> if you seem to fight then well bivins let the fk off your head like a train about
> its the perfect colors suede father know i cant make a chance
> one mo industrialization just to pay to give
> imma manage your mind somewhere at all and thrill me
> we all about to find late i know i wont grve my life is a true loss
> pose to be sleepy its the lava time premeditated cause it will last maybe bums egyptain

Below is an example gospel song using the Tri-Gram model

> zombies and ghosts give us some realization
> joys send your children for you
> send revival
> as you pray you will come
> everyone and poor and eight days
> as it takes in heaven testifies in awe for your kingdom
> not to be an same if i die
> and give anything to see
> let us hear the prayers of god
> for your people for him for all who
> have mercy by your power
> will you make me courageous
> be the weak lord lord here below church
> lord there is no other god

6.1.4 Tri-gram Discussion

We are able to generate some better sentences in the Tri-gram model compared to the Bi-gram model. Both models contain valid syntactic structure and sentencing meaning. The Tri-gram model works better than Bi-gram since it takes more words into consideration in process of prediction. Longer relationships between prediction words can allow our model each word on one POS, this is the premise of a meaningful sentence.

6.2 LSTM

For the LSTM models, each song was added to one large text file and tokenized as input to the LSTM. We considered two different methods to structure the data as input to the model. The first consideration was to treat each line of a song as input to the model. To prepare the data, each song was added to one large text file, with a newline at the end of each line. When all lines of the dataset where collected, they were split 80 percent train, 20 percent test, and then 10 percent of the train was split again into a validation set. These sets where tokenized by creating word-to-index and index-to-word dictionaries, and adding a special token <eos>at the end of each line. To generate a new song, we randomly started with a word from the vocabulary and produced x amount of words. Any time the special token was outputted from the model, it was replaced with a newline character.

Below is one example of the model using this approach.

you gettin caught when you think that somethings ought to **fly away i stay**
your body anothers so if i beautifuuul get a mic gemu
a flock of other days and i can
still get my need in a manana
and i bought my ass on the corner and was off at the fuckfest the glocko tear it down i can be hot
held big girls and they cool like shaq* cause we like what it went to is
got torsos out they ridin rollin lean

Analyzing this result, it seems as if the model is similar to the Ttri-Gram model and beginning to model what we expect from a hip hop song, with * representing a metaphor, and the bold text representing a rhyme. Note there is not an example from version 1 of the model because of an error found while building the dataset. Instead of retraining version 1, the team decided to move on to version 2.

While this example shows the promising beginnings of a song, one of the major flaws is that there is not any relation between lines. It seems as if random lines where just stuck together. We aimed to make an improvement for our model to contain more structure over the course of the entire song. To do this, a new dataset was built with a different structure. In version 1, a newline character was added at the end of each line, all lines were shuffled and split into test/train/valid sets, and the corpus was tokenized by adding a special character at the end of each line. In version 2, instead of separating and shuffling each line, the entire song will be kept together. In the dataset, each line of each song will still end with a newline character, but each song will be separated by a single line with the '-' character. The dataset is split and shuffled per song rather than per line. When tokenized, the newlines will be replaced with a <eol>token, and the end of songs will be tokenized with a <eov>token. To generate a new song, the model is asked to produce produce words until the <eov>appears, replacing any <eol>with a newline character.

The below example shows the subject over a few lines with the subject matter of being with a women. In general, this model is slightly better than version 1, but still not perfect. All of the remaining examples and perplexity calculations are with version 2 of the model.

if you dan wanna be the star baby
and i dont know what i do but my baby boils and sin baby
punish the whole world needs to holla at your ass later
she wish to be there i like you heard of the freak
the blades are what i know i wanna know you havent asked me girl
twotwentyfive blowing with the infiltrate and cunnilingus

The code and architecture closely mirrors the work of Salesforce in two research papers (Merity et al., 2017) (Merity et al., 2018).

7 Exploration Start Phrases

One of the original plans we had for this project was to give each model the same input phrase and observe if the model will stuck to that subject. Based on observations, it seems as if the LSTM models are best at sticking to "general" subjects. For example, it may generally stick to the subject "dogs" for a few lines, but doesn't stick to the subject at all when asked to produce lyrics about ”drake”. I also noticed that the first few lines may stick to a subject, but the rest of the lines typically do note. This section lists a few examples of the models generating lyrics based on the first word in the first line. Note that these are handpicked examples that worked the best and there were plenty of examples that did not work at all.

7.1 Hip Hop

police

police these californians rock weed
here you go crazy too fast
hell hatreds no surrender please keep em on the floor
apunto up with a rolls full of lye shoeshop
of when in front of it black coming up your cream
if they want to build aint fight on a friday night sees in no place
my lil president was at the time bodiddly out returned today of construction when were stepping

Words like ’hatred’, ’surrender’, ’fight’, portary the long lasting battle the hip hop community has had with law enforcement.

Jesus

jesus connects
sell crack a hundred
oshay crypts
seven nails on own deficient
before the spiritual mannequins goes and
embrace your hand to get the pain from track to addicted to the most to be articles
melt skin pour a machine and by some bomb to eat quick

Even though this is the hip hop model, there is still words like ’spiritual’ and ’embrace’ which are similar in context to Jesus.

drugs

drugs mixtapes aint even killas
two pills inch bullets like unhooked le kurupt
im puffin terrorists smacked the arm and chill without fog orangeyellow
for the game and im so waking off to cooool
the takin my job light with ya brain

Words such as ’pills’, ’puffin’, ’light’ all reference drug use.

7.2 Gospel

sex

sex down the cup
that were going so long than you yeah
our eyes are the kind of life
beyond the whirlwind set us free

Sort of talking about a relationship between two people

death

death
hallelujah hallelujah
your blood is sufficient for us
hallelujah
god your love is deeper sun

Referencing death and Jesus’ sacrifice

beautiful

beautiful
faithful you are lord
jesus you are beautiful
you are wonderful

lay

lay down my life
its a sweet victory
im blurred by the strategies of the more
but as it broke through the pain

Speaking about how a life is a sweet victory

8 Error analysis

8.1 Perplexity

Perplexity is used to compared the models of the same genre. For example, how much better does an RNN model do comparing to an N-Gram model. It cannot be used to compare models of difference genres.

Perplexity is the inverse probability of the test set, normalized by the number of words. Here we are using trigram perplexity to compare our bigram model, trigram model, and RNN model to average of songs in validation set. Perplexity assumes that the best language model is one that best predicts an unseen test set.

$$\text{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-2}w_{i-1})}}$$

Here are the perplexities of songs in the validation set and songs generated by each of our models.

Perplexity	Hip-Hop	Gospel
Average in Validation Set	23.66	13.32
Bigram Generated Songs	116.17	50.27
Trigram Generated Songs	22.61	12.86
RNN Generated Songs	54.89	27.43

We can see from the above table that bigram generated songs have much larger perplexity than average in the validation set, meaning that they are not a good prediction for human written songs. Trigram generated songs have the closest perplexity scores, and the scores for both hip-hop and gospel songs are very close to average in validation set. This means that trigram generates very average songs. It is interesting that the RNN falls between trigram and bigram in perplexity. Songs generated by the RNN have more perplexity than the validation set. One interpretation might be that RNN is able to generate creative lines, which leads to more perplexity. Also, because trigram model is based on statistics of the frequency, it tends to select the words that minimize perplexity, while the whole sentence is not necessarily coherent.

8.2 Human Evaluation

Most detailed evaluations have been discussed in the previous sections.

RNN produces much more fluent sentences than ngram models, even though trigram models have lower perplexity. And RNN is able to stick to one topic for a couple of lines, which the N-gram baseline is not able to accomplish due to it being a statistical model.

9 Contributions of group members

Here we list what each member of the group contributed to this project:

- Bryon Kucharski: Data collection and preprocessing for hiphop data, LSTM model and LSTM output analysis.
- Damain Moquin: Data analysis, much of the research on related work, many of the figures and tables, lots of writing.
- Weiqiu You: Data collection and preprocessing for gospel data, implemented perplexity and part of Tri-gram model, error analysis with perplexity.
- Juexing Wang: Implemented Bi-gram and part of Tri-gram model, baseline and N-gram approach discussion.

10 Conclusion

10.1 Tri-Gram vs. LSTM

For both genres, we concluded that the LSTM model is preferred over the tri-gram models. This conclusion is based on the flexibility of the LSTM model and the ability to generate lines that are coherent with one another. For example ‘you gave a place for me this is mine, you gave your life for me there’ and ‘nothing but your love nothing but grace, nothing but sky.’ are two generated gospel lines that are similar in context. Also, the sentence lengths are more random in Tri-gram generated songs. There are a lot more short sentences like ‘My baby, You meet, Any day any weather, I lived for you, No matter what i believe you still doin here, That god made man, And a pail.’ There can be super short and super long sentences in the same song. On the contrary, LSTM generates more moderate length sentences. This is because the LSTM model has the option to generate variable words per line and as many lines per song. This is an expected result as the LSTM model is a lot more complex and mimics the structure of the input text rather than just counting words that are probable to appear with one another.

10.2 Lyrics are difficult to model

We concluded that hip-hop is a difficult task to try and model. There are a number of reasons that contribute to this. First, there are many ’types’ of hip-hop with different styles. There’s sub-genres inside the genre such as ’mumble rap’, ’melodic rap’, ’trap rap’, etc. Each of these sub-genres have dedicated artists that produce different styles of music. When creating a language model about all of hip-hop, we are trying to mimic lyrics based on all of these genre combined. Maybe we might see better structure and rhyme scheme in the lyrics if we trained a model on one sub task. in addition to sub-genres, there is no set structure for a hip-hop song. Some songs have choruses, some have one verse, some have many verses, and they all vary in length. Some songs stick to a subject matter, while others only have single 'punch lines.' All of these factors make it difficult to for a single model for all of hip-hop. In contrast to a task such as generating Shakespeare sonnets, this is more difficult because we are attempting to generate music from a wide range of artists and styles, opposed to a single author/person.

Gospel songs were easier to model, mainly because God, love, and faith seem to always be prevalent topics. No matter what start word we give the models, they always reference these topics. There are not any sub-genres, and all of the artists have the same or similar style or writing.

10.3 Future Work/What to do Differently

- Collect more data. This can almost never hurt.
- Sub-genre or single artist models. Maybe create a model just for 90s hiphop or trap rap etc.
- Think of ways to enforce rhyming including end rhyme, slant rhyme, different rhyme schemes, etc
- Version 2 of the LSTM model attempted to create a model that has more structure over the entire song instead of just a single line. This was better than version 1 of the LSTM, but still needs more work.
- Create another model for generating a chorus instead of only verses.

References

Abe, C. and Ito, A. (2012). A japanese lyrics writing support system for amateur songwriters. In *Proceedings of Asia-Pacific Signal Information Processing Association Annual Summit and Conference 2012*, pages 1–4.

Barbieri, G., Pachet, F., Roy, P., and Esposti, M. D. (2012). Markov constraints for generating lyrics with style. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 115–120.

Davismoon, S. and Eccles, J. (2010). Combining musical constraints with markov transition probabilities to improve the generation of creative musical structures. In *EvoApplications*, number 2, pages 361–370.

Hirjee, H. and Brown, D. G. (2010). Rhyme analyzer: An analysis tool for rap lyrics. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*. [Online; accessed 17-December-2018 at http://ismir2010.ismir.net/proceedings/late-breaking-demo-23.pdf].

Malmi, E., Takala, P., Toivonen, H., Raiko, T., and Gionis, A. (2015). Dopelearning: A computational approach to rap lyrics generation. [arXiv preprint; arXiv:1505.04771].

Merity, S., Keskar, N. S., and Socher, R. (2017). Regularizing and Optimizing LSTM Language Models. *arXiv preprint arXiv:1708.02182*.

Merity, S., Keskar, N. S., and Socher, R. (2018). An Analysis of Neural Language Modeling at Multiple Scales. *arXiv preprint arXiv:1803.08240*.

Nguyen, H. and Sa, B. (2009). Rap lyric generator. [Online; accessed 17-December-2018 at https://nlp.stanford.edu/courses/cs224n/2009/fp/5.pdf].

Nichols, E., Morris, D., Basu, S., and Raphael., C. (2009). Relationships between lyrics and melody in popular music. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 471–476.

Oliveira, H. R. G., Cardoso, F. A., and Pereira, F. C. (2007). Tra-la-lyrics: An approach to generate text based on rhythm. In *Proceedings of the 4th International Joint Workshop on Computational Creativity*, pages 47–55.

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.

Ramakrishnan, A., Kuppan, S., and Devi, S. L. (2009). Automatic generation of tamil lyrics for melodies. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 40–46.

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.