

Rap Lyric Generator

Hieu Nguyen, Brian Sa

June 4, 2009

1 Research Question

Writer's block can be a real pain for lyricists when composing their song lyrics. Some say it's because it is pretty hard to come up with lyrics that are clever but also flow with the rest of the song. We wanted to tackle this problem by using our own song lyric generator that utilizes some Natural Language Generation techniques. In the general case, our lyric generator takes a corpus of song lyrics and outputs a song based on the words from the corpus. It also has the ability to produce lines that emulate song structure (rhyming and syllables) and lines that are tied to a specific theme. Using the ideas produced by our song lyric generator, we hope to provide lyricists with some inspiration for producing an awesome song.

We chose to use only rap lyrics for our lyric corpus because we thought the language used in rap lyrics were very specific to its domain, and thus interesting to read. Also, the lyrics often have a similar structure (similar word length per line and similar rhyming schemes). Our lyric generator can be applied to any other type of lyric, such as rock or pop, or even to poems that have some structure and rhyming.

2 Related Work

Natural Language Generation is a rapidly evolving field of natural language processing. It can be used in fun hobby projects such as chat-bots and lyric generators, or it can have applications that would aid a larger range of people. There has been work in automatically generating easy-to-read summaries of financial, medical, or any other sort of data. An interesting application was the STOP Project, created by Reiter, et al. Based on some input data about smoking history, the system produces a brochure that tries to get the user to quit smoking, fine-tuned to the user's input data. The process is divided into three steps: planning (producing content), microplanning (adding punctuation and whitespace), and realization (producing the brochure). The system did produce readable and quite persuasive output. But results showed that the tailored brochures were no more effective than the default non-tailored brochures.

Work in Natural Language Generation revolves around creating systems that produce text that makes sense in content, grammar, lexical choice, and overall flow. The systems also need to produce output that is non-repetitive, so they need to do things like combine short sentences with the same subject. In general, Natural Language Generation systems need to trick readers into thinking that the generated text was actually written by a human.

```
--talking--
Lets get it on every time
Holler out "Your mine"

[Chorus] [10sion not singing]
And I say "a yia yia yia" --singing==
Let's get it on every time
Holler out "Your mine"
And I say "Oh oo oh oo oh oh oh oh" --singing==
So if you willin' you wit it then we can spend time
And I say "a yia yia yia" --singing==
```

Figure 1: Excerpt from 10sion's "Let's Get It On"

```
Chorus:
Everybody light your Vega,
everybody light your Vega,
everybody smoke, woo hoooo (2x)

Chorus

Now first let's call for the motherfuckin indo
Pull out your crutch and put away your pistol
<Rest of verse>
```

Figure 2: Excerpt from 11/5's "Garcia Vegas"

3 Implementation

3.1 Data

3.1.1 Rap Lyrics

We crawled a hip-hop lyrics site (www.ohhla.com) and pulled in about 40,000 lyrics from artists ranging from 2pac to Zion I, putting them into a MySQL database. We then preprocessed a subset of those lyrics by removing the header, removing unnecessary punctuation and whitespace, and lowercasing all the alphabet characters. Finally, we split the content of the lyrics into **chorus** and **verse** flatfiles. This was actually not a trivial task. The lyrics from the site were in various formats and used different headers, so it was difficult to tell where chorus sections began and ended.

As seen in Figures 1 and 2, the two lyrics use different formatting for Chorus headers. Also, as in "Garcia Vegas", it was hard to tell whether a section actually corresponded to the chorus, or if the word **Chorus** was just used to indicate a repeat of the chorus. This occurred in several other songs. We solved this by using a state machine as we were parsing the lyrics line-by-line to keep track of which section we were in. We had to manually create the transition rules for the state machine. For example, if we saw **Chorus** then a blank line, we would assume that the next section is actually the verse.

Each flatfile contains a single lyrical line (which we will define as a "sentence") per line in the file. Our language model uses this data to train.

3.1.2 Rhyming Words

We used a rhyming database (rhyme.sourceforge.net) to produce words that rhymed with a given input word. The rhymer's default usage is through command-line, and although this produced results, we eventually decided to create flatfiles of all `word` \rightarrow `rhyme` possibilities for all the words in our `chorus` and our `verse` corpora. These files also included the syllable count of the words. When our lyric generator is loaded, it loads all of the rhyme flatfiles into memory.

3.2 Language Model

Our rap generator uses two language models: one that produces the chorus, and one that produces the verse. They are essentially the same model, except trained on different corpora.

We originally started out with a linear-interpolated Trigram Model that weights the scores of absolute-discounted unigram, bigram, and trigram models according to hand-set weights. Although this produced decent results, there was a general lack of flow in the sentences because our model only looked at a 2-word history to produce the next word. Here is an example line from our Trigram model:

```
comfort pigeons feeble need me i don't park there's a knot
```

We then created a linear-interpolated Quadgram Model that weights the scores of absolute-discounted unigram, bigram, trigram, and quadgram models according to hand-set weights. This produced much better results, like this example:

```
what you know you gotta love it new york city
```

3.3 Sentence Generation

For each section in the song (chorus or verse) we generate a set number of lines using the corresponding language model. For each line we generate, we actually generate a certain number of candidate lines (K , default = 30) from the model, and rank them according to a score. Then we pick the sentence with the best score, and repeat the process to generate all the lines in that section.

This score comprises of several different metrics:

1. The log probability of the sentence from our language model, divided by sentence length
2. The log probability of the sentence length
3. The sum of logs of TFICF (term frequency-inverse corpus frequency) of each word in the sentence
4. Whether the last word of the line rhymed with the last word of the previous line
5. Whether the last word of the line rhymed with another word in the sentence
6. Whether the last word of the line had the same number of syllables as the last word of the previous line

Notes for each metric:

1. We want to make sure that the generated sentence actually fit the model we were generating from, so we calculate the sentence probability based on the model. We had to divide the log probability of the sentence by sentence length because longer sentences have lower probabilities due to the fact that more word probabilities are being multiplied together. This takes out the bias toward shorter sentences, so then we can utilize our second metric to score based on sentence length.

2. We want our sentences to emulate the length of the sentences in rap lyrics, so we tried to account for sentence length in our score. The most common sentence length was 9 for verses, and 8 for choruses.
3. To include thematic information from a given input song, we generate TFICFs for each word in our song. We define TFICF as the probability of the word in the song divided by the probability of the word in the corpus, which corresponds to how important and specific the word is to that particular song. If a word in our generated sentence is not in our song, we defined TFICF as the minimum TFICF squared. So our score metric is just the sum of the logs of these TFICFs for each word in the generated sentence.

Finally, we piece together each section in the song according to some predefined song structure (i.e. verse-chorus-verse-chorus).

4 Testing and Results

4.1 Rap Quality

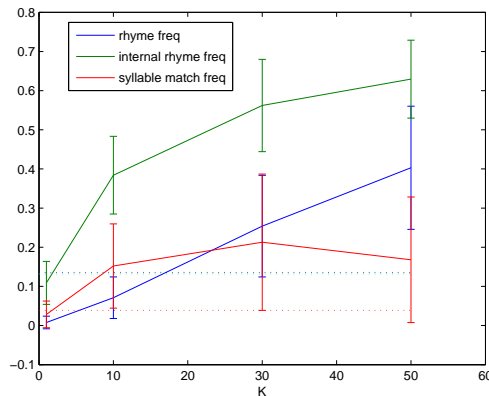


Figure 3: Average rap quality per song as a function of K (number of sentences generated per line)

Each line in the rap is generated by generating K lines using our language model and then evaluating them for end rhyme with the previous line, internal rhyme, and matching syllable count to the last word in the previous line. This was our measure of quality, and as seen in Figure 3 it goes up as K increases. The means are plotted with error bars that indicate the standard deviation over 300 generated songs for each K. The dotted lines are the respective rhyme frequency, internal rhyme frequency, and syllable matching frequency in the training corpus. Our generated raps surpass the baseline which indicates that there are other hidden factors we are not taking into account when assessing rap quality. Figure 4 shows how average rating per sentence increases as K increases, but is probably inflated.

4.2 Example Output

The real joy of our Rap Generator is actually reading the outputted lyrics and seeing if they make sense, and if it is possible that they could have been written by a human. So we will examine two sample outputs: one generated using an input song, and one generated without using an input song.

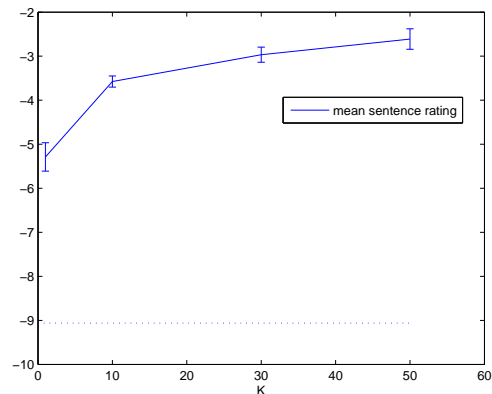


Figure 4: Average rating per line as a function of K

i like getting head 'cause it's so convenient huh
you can do it any time you don't have to beat it
you can get it in the car or even in the park yeah
but most head-hunters go out after dark true
there's nothing like a pretty hoe on her knees
suckin' my d yeaah. and lickin' my b's
we don't have to take our clothes off to bust a nut
when i pull out my dick biiiitch pucker up
and gimme some
coolin' in a club on a saturday night
gettin' fucked up with the boys and feelin' all right yeah
when i saw this bitch who had to be a winner
and the only thing on my mind was to run up in her
so i got her kind of tipsy with some sex on the beach
then the bitch got hot and she wanted to eat
so for me to get over took her straight to the bed what
she got on her knees and gave some good hot head yeah
hot head hoes some white some niggeroes
but i like the ones who suck toes and assholes
with tongues like razors that cut when she licks ooh
how can i fuck you with a skinless dick ha ha ha ha
you take pride in suckin' a good dick
and after i nut bitch you better not spit ha ha ha ha
you're a dirt dobbler a goop gobbler
you'll fuck satan for the righteous dollar
so gimme some gimme some

Figure 5: Original Song: 2 Live Crew's "Some Hot Head"

Verse:

the bigger the butts the tighter the clothes
ha ha ha ha
try them and you may i say
what more could i say i wouldn't be here today
when you reach the pearly gates how you gon' explain
who do you run to
and chuck d and the crew
drinkin on gin smokin on blunts and it's on
but when it's on it's on
while she hot and horny all up on me
run up on me
and if a nigga get some head
all hoes suck dick
take the good with the bad and the bad with the good
what the lord have to take our clothes off to bust a nut
that we have don't way out

Chorus:

can a nigga get some to go yeah baby
she got it she got it she got it
i do my thang in the club
you can do it

Verse:

cause i eat up tracks like hannibal and dahmer
so the ones who do
be true to you
feelin' all right on a party night
when it's time to go on a bus
she got on her knees and gave some good hot head yeah
you and me got to be
in the telly with to whores a benz with to doors
then take it to the limit take the shit to me
fuck around and got it twisted you can get it
a diamond car with a bar and a spa
huh reach for my drink and for a second
it be astoundin formatix around it it paint drank down it
get it get it get it girl
then we can use the rabbit all over your cat
get yaself a beer get on the floor

Chorus:

can a nigga get some to go yeah baby
she got it she got it she got it
i do my thang in the club
you can do it

Figure 6: Our Rap Generator's output based on "Some Hot Head"

Verse:

my nasty new street slugger my heat seeks suckers
now i'm a pimp you a player
i i'll rob boys ii men like i'm michael bivins
if i'm from southside jamaica queens nigga ya'heard me
i and to the punk police can't fade me
so pussy claat bwoy ya nah wanna ruf wif me
i'm in a party where some suckers was at
it's a fuck-nigga from atlanta named after me what
and i could touch y'all haters from a mile away
on not braggin' i'm bad and i could only get better
a when they say 2 live your mama gets worried
we both sides begin anew the quest for peace
now i won't deny it i'm a straight ridah
we turn to spurn desire - that all
you from the cradle to the grave
they ha ha ha ha ha ha ha

Chorus:

i all i wanna do is spend some time with you
uh i hope you wear a vest souljas touchin' you touchin' you
but it's alright with me if it's alright with you
nigga i woke up and screamed fuck the world

Verse:

but there's no escape nah i ain't ready to die
and she said she wants to come home and
an see mama raised me to be a man
you all you sucka duck rappers your era is through
now i heard you screamin our name whatup with you
it's yeah all the homies that i call my crew
tony 2 li 2 li 2 live crew
when i'm the type to go spark metal in
you even a smooth criminal one day must get caught
say drop the drums here it comes only got
see i won't deny it i'm a straight ridah
i got semi-autos to put holes in niggaz tryina play me
i look to my future cause my past is all behind me
yeah see the cross on my neck that just might freeze me
shine now if greed come between me and my man d
well they say they wanna question me

Chorus:

i all i wanna do is spend some time with you
uh i hope you wear a vest souljas touchin' you touchin' you
but it's alright with me if it's alright with you
nigga i woke up and screamed fuck the world

Figure 7: Our Rap Generator's output based on no input song

The lyrics presented in Figure 5 revolve around the theme of receiving oral sex, alcohol, and going to the club. Thus, words like “club” and “bust” have relatively high TFICF scores (TFICF=1.83e-4 and TFICF=1.67e-4) than other non-related words (TFICF=4.24e-7). The lyrics presented in Figure 6 also generally revolve around the themes of sex and partying.

We also noticed that rhymes showed up very often, such as within the line

feelin' all right on a party night

and between lines

so the ones who do

and

be true to you

One can compare these lyrics to the lyrics from the previous version of our Rap Generator that did not take the input song into account (Figure 7). From reading the lyrics, one can see places where many themes were mashed haphazardly into one song. A shining example of this is in the chorus, where our Rap Generator seems to switch from a romantic tone in the first line, to a warning-like tone in the second line, back to a romantic tone in the third, and finally to an aggressive tone in the fourth.

5 Alternative Methods

5.1 Pivot Word from Input Song

In order to incorporate input song theme information, we first tried a different method than the one we described in “Implementation”. We tried generating sentences by picking words from a given input song, and generating forward and backward from that “pivot” word. This was an attempt to include thematic information from the input song by using some of its vocabulary. However, each model is trained on data that has a certain average sentence length, which is equal to the sentence length that we generally desire. And since we are generating fragments from two models and piecing them together to form a sentence, the desired fragment length is half of our desired sentence length. Since the desired fragment length is much smaller than the average sentence length in our corpus, the fragments generated were not too great.

Here is an example of a sentence generated by this method (using a Trigram model, where “0-5” was our pivot word):

all the game with our hearts remain larock 0-5 beaming scoop

The sentence length is fine, but the sentence seems a little forced to include the “0-5” so the words don’t mesh well with each other.

6 Discussion and Future Work

6.1 Utilizing Sentence Structure Parse

Although our Quadgram model produces sentences that are usually “grammatically correct” according to rap lyrics in general, we do not actually utilize a model that models grammatical sentence structure. One thing we could have done is use a parser, such as the Stanford Parser, to create parse trees from sentences in our corpus. Because the default training data for the Parser is from a

different domain, we would probably have to manually create several “gold standard” parses based on rap lyrics.

Once we generate these parse trees, we can count the **part-of-speech** \rightarrow **word** pairs to create probabilities of a word given a part of speech. We can also count the **parent-node** \rightarrow **child-node-list** pairs. This can be useful to create trees of our own. Starting at the root, we can recursively build a tree downward randomly based on our calculated probabilities until we end up with a tree with only parts of speech at the bottom. Then, using this “madlib” structure, we can generate a word for each part of speech. This, we believe, will generate sentences with generally good grammar and good vocabulary.

6.2 Clustering to Create Themes

One problem with our generator is that themes seem to switch from line to line. For example, one line may be about carrying weaponry, and then the next may be about going on a bike ride. Given a corpus of lyrics, we think it may be useful to cluster songs, or even lines, based on vocabulary. For each cluster, we can build a model that trains on only lines from that cluster. Then, for song generation, we can just use one of these themed models to generate lines that relate to a specific theme.

7 References

- The Original Hip-Hop Lyrics Archive: <http://www.ohhla.com/>
- Rhyming Dictionary: <http://rhyme.sourceforge.net/index.html>
- E Reiter, R Robertson, and LM Osman (2003). Lessons from a Failure: Generating Tailored Smoking Cessation Letters. *Artificial Intelligence* 144:41-58.