Rap Overflow: Using Seq2Seq Fine-tuning for Rap Lyric Annotations

Jonathan Tran
University of California, Berkeley
bk_pbjonmt@berkeley.edu

Abstract

The focus of applying LLM's to music has been largely focused on lyric classification, lyric summarization, and lyric generation. However, one underexplored area is applying LLM's to read between the lines and to understand the hidden meaning of song lyrics, particularly for rap music. Since no studies comparing multiple LLMs using seq2seq exist, I compared the performance of T5, Pegasus, and BART to determine which LLM is most suitable for annotating rap lyrics using seq2seq encoder-decoder architecture. Using an ensemble of evaluation metrics, BART was superior to T5 and Pegasus in generating better annotations, displaying faster convergence, and exhibiting higher performance even with fewer data used for fine-tuning.

1 Introduction

Understanding the meaning of song lyrics can be challenging to humans and machines alike. In some cases, lyrics are straightforward, but other times, artists make use of ambiguity, symbolism, and wordplay that make lyrics challenging to interpret. This is especially true for a genre like rap which consists of slant rhymes, abnormal sentence structure, and figurative language.

The focus of applying LLM's to music has been largely focused on lyric classification, lyric summarization, and lyric generation. These studies are focused on condensing information or replicating information.

However, one area that is underexplored is reading between lines to understand hidden meanings, which combines both translation and generation. While some have made attempts to annotate a generalized corpus of songs, rap lyrics annotation has been a neglected topic. Rap songs are particularly hard to interpret relative to other songs especially because of rappers' use of slang and the cultural knowledge required for lyrical interpretation. As a result, existing datasets are not suitable for this task and they do not offer the utility of vote counts and contributor scores for researchers to segment data. In the process of the research, I have created the rap genius dataset which can now be accessed on hugging face.

Previous studies related to song lyric annotation suggest that a seq2seq fine tuned LLM can be successful in this task of lyric annotation, but there are no existing studies comparing multiple LLMs using seq2seq. In this study, I will compare the performance of T5, Pegasus, and BART to determine which LLM is most suitable for annotating rap lyrics using seq2seq encoder-decoder architecture.

2 Background

Most work regarding song lyrics has focused on either the generation or summary of songs more than interpretation. For example, Fell et al.(2019)[2] leverages audio thumbnailing to summarize song lyrics. Nikilov et al. (2020)[3] generates rap lyrics using denoising autoencoders. However, for lyric annotation, there have only been a handful of studies that attempt to explain the underlying meanings of song lyrics (Sterckx et al., 2017[4], Ventura & Toker[5], 2022, Li et al., 2023)[1]. Sterckx et al., 2017 introduced that task of automated lyric annotation (ALA) and created their own Genius Annotation dataset. They also coined the distinction between context independent (CI) annotations and context sensitive (CS) annotations. They achieved some success using SMT and Seq2Seq models and suggested that Seq2Seq models "demonstrated potential to generate more fluent and informative text, dissimilar to the lyric" (Sterckx et al., 2017). Ventura

& Toker utilized GPT2 (decoder only) and T5 (encoder-decoder) and experimented with using different prompts that included metadata to improve results. Their work demonstrated that using the artists' names and the names of the songs as part of the prompt can improve model performance. As a result of the study, they produced the TRBLLmaker dataset and their GPT2 model outperformed T5. Li et al., 2023 improved this approach of using T5 by applying Named Entity recognition and utilizing api calls to GPT 3 for information retrieval. Using this method, Li et al., 2023 was able to produce a model that outperformed the models of Ventura & Toker. To further advance this line of research, I have two goals. The primary goal is to study which seq2seq models and parameters are most suitable for the task of ALA after fine tuning. A secondary goal is to understand how data quality and quantity impact the fine tuning process. Furthermore, previous studies have attempted to utilize a non-genre specific corpus. My study will use a genre specific corpus as well as consider annotation quality to attempt achieving better results. Additionally, the previous studies have not attempted to utilize other modern LLMs with the potential for seq2seq performance such as Pegasus and BART.

3 Methods

3.1 Dataset

Because existing data contained song lyrics and annotations from a variety of genres, I opted to scrape my own dataset from Genis.com called the Rapgenius dataset which has been published on Huggingface.com. This dataset comprises lyrics and annotation pairs, obtained by scraping lyrics from 150 rappers. In total, I gathered 230,719 lyric-annotation pairs this dataset is superior to existing datasets because it includes information regarding the annotator IQ, the number of votes for each annotation, and additional song metadata*explain*. Using these additional fields, I was able to curate 118,388 annotations with annotator IQ >= 5400 or the presence of 83 or more votes on the annotation to test performance on high quality annotations. Users can navigate to a specific part of a song, view existing annotations, contribute their own, and vote on the quality of existing annotations. The collaborative annotation process involved users earning IQ points for producing high-quality annotations in a gamified environment, similar to collaborative efforts seen in platforms like Wikipedia. Our dataset uses the annotation for each set of lyrics.

3.2 Experiment Design and Implementation

The general problem to solve is which LLM is suitable for fine tuning to to annotate rap lyrics and how can we fine tune it optimally. The primary questions for this study are:

1. Which LLM performs best for seq2seq modeling?

2. How do quantity and quality of training samples impact fine tuning results?

3. How can training hyper parameters improve performance?

3.2.1 Task 1: Model Comparison

The models I used are T5, Pegasus, and BART. Rather than trying to optimize parameter tuning at training and generation, I opted to prioritize the simpler approach of using the out-of-the-box configurations for each model and the same generation parameters when decoding. I compared each model with several data set sizes to see if one model can outperform another model as the volume of fine tuning data increases. While all three models have demonstrated success with Seq2Seq tasks, my intuition was that the original data used for pretraining predispositions one model to perform better than the others. The initial training consisted of 60 tokens due to resource constraints in Colab, all other training settings including the loss function are based on the defaults from the HuggingFace Seq2Seq Trainer. Each model was run for 10 epochs, but some stopped early because of convergence. To improve consistency, the generation parameters used in each model for tasks 1 and 2 were set using the following parameters: no repeat_ngram_size=4, max_length = 100, num_beams = 4.

3.2.2 Task 2: Annotation Quality Comparison

For this task, I checked whether higher quality data could improve the performance of annota-

tion generation by fine tuning each model on a "curated" *footnote definition* data set. Due to Colab resource constraints, the training data set sizes for this task was 16k each. I trained each model on one data set that was curated and one data set that was a mix of curated and uncurated annotations. Although I already took the best annotation for each lyric, I was curious if training on a set of more highly rated annotations will yield better results. Assuming there are the same number of training examples, the purpose of this task was to understand whether the more highly rated annotations would result in higher performance.

3.2.3 Task 3: Training Parameter Tuning

For this task I checked to see the extent performance increased with training parameters. In order to contain the scope of the project, I excluded model generation parameters. Instead,I took the best performing model and increased the max_length parameter to see if its performance improved final results. I hypothesized that at generation, the model would benefit from additional context from training.

Baseline (T5) The baseline model was a finetune, pre-trained T5 model from HuggingFace on my rap lyric dataset. I wanted to pick a baseline that was based on previous research. Furthermore, Sterck et al suggested that seq2seq could provide more robust annotations than using Neural Machine Translations (NMT) and Li et al demonstrated strong improvement by using T5 with additional enhancements.

Pegasus Pegasus is a pre-trained sequence-to-sequence model developed by Google Research for abstractive summarization. Ultimately, I decided to use Pegasus/Newsroom after testing against other versions of the model. Pegasus has not been used for rap annotations by any previous study.

BART BART is a pre-trained sequence-to-sequence model introduced by Facebook AI Research (FAIR). In initial testing, BART-large outperformed both T5 and Pegasus, so I opted to keep BART-large rather than testing additional versions of BART. BART has not been used for rap annotations by any previous study.

3.3 Evaluation Metrics

For evaluation metrics, I used an equal weighted ensemble method consisting of BLEU, ROUGE-F1, METEOR, and Cosine Similarity.

BLEU (Bilingual Evaluation Understudy) was used because of its wide adoption for machine translation tasks and measurement of ngram precision.

ROUGE-F1 (Recall-Oriented Understudy for Gisting Evaluation) was used because previous studies have opted to use this metric. Rouge is also case insensitive which allows for maintaining the original case of the annotation text while still being able to compare it to the prediction. ROUGE F1 is a combination of precision and recall, providing a balanced assessment of content overlap between the generated and reference sequences.

METEOR (Metric for Evaluation of Translation with Explicit ORdering) was be used because previous studies have opted to use this metric. It is based on the harmonic mean of precision and recall and checks for exact word matching, stemming, and synonymy matching.

Cosine Similarity was be used because previous studies have opted to use this metric. This metric is particularly good at capturing semantic similarity between sequences while also being robust to synonyms and variations. Semantic similarity is important because we want to ensure that the annotations generated have a similar meaning to the actual annotations. To simplify evaluations in the paper, I will reference the ensemble score, and I will consider specific components if relevant.

4 Results and discussion

4.1 Results : Model Comparison

Both Bart and Pegasus outperform the T5 baseline in evaluation metrics. Considering the ensemble scores for each model trained on the maximum data set size, Bart scores: 0.183, Pegasus scores: 0.169, T5 scores: 0.153. Interestingly, even when training on the minimum number of training samples BART (0.176) outperforms both T5 and Pegasus using the maximum number of training samples. See chart with performance. Increasing the volume of training examples yielded similar results. Given the same number of training examples, Bart outperforms both T5 and Pegasus.

Model	16k	80k	160k
Bart	0.176	0.174	0.183
Pegasus	0.167	0.168	0.169
T5	0.146	0.155	0.153

Table 1: Considering multiple training set sizes, Bart has the best performance with respect to the ensemble score. The full list of experiments is listed in the appendix

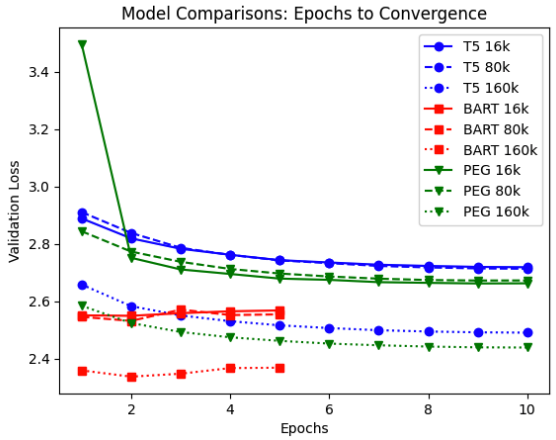

Figure 1: Bart seems to converge after the second or third epoch in comparison to other models.

In the process of training the models, I noticed an interesting loss pattern that seems to suggest that the Bart model sees earlier convergence while fine tuning. This finding was particularly exciting because tuning efficiency and computation resources were not originally a focus of the project. Given the results of the first experiment, BART seems to not only have the highest performance, but BART also may be more likely to result in lower computational cost due to lower training times for the same data set.

The three models vary greatly in their depth of analysis. First, T5 seems to understand that "Beef rap" is a slang term, but cannot explain the meaning. T5 also stops the text generation early without finishing the sentence. On the other hand, Pegasus is able to generate complete sentences (see examples in Appendix that demonstrates Pegasus's high competency at generating complete sentences across prompts). More importantly, Pegasus contextualizes "Beef rap" as taking place between multiple individuals and provides a definition of teeth caps. However, Pegasus does not detect the cultural undercurrents of rap culture on "Beef rap" and incorrectly assumes that the use of teeth caps is for protection. Lastly, BART is able to not only provide a more detailed and more accurate description of "Beef rap" than either model but also infuses rap culture in its lyrical analysis.

Bart identifies that "Beef rap" takes place between rappers who speak poorly of each other behind each others' back. It also indicates not only the potential for violence associated with "Beef rap" but also the rationale behind the violence as an expression of the rappers' anger and frustrations. Although the original lyrics only refer to "Beef rap" and do not contain any profanity, Bart generates an annotation with profanity (which I manually replaced with "****"), as if the annotations themselves mirror the rappers' anger and frustration. Even the lyrics themselves do not reference any names, Bart uncovers a hidden, plausible interpretation by referencing Action Bronson. Action Bronson has been featured in the news[1] for Beef rap. Here, Bart reads between the lines and presents a hidden meaning of the rap lyrics, which demonstrates the great potential in utilizing machine learning models to uncover the complex nuances of song lyrics.

4.2 Results: Annotation Quality Comparison

For the second task, curation did not seem to improve results based on my experiments and example data. T5 seems to see performance increases, but Pegasus and Bart both see performance decreases.

One reason why my results may be impacted is an imbalance of annotations by rappers. More popular rappers tend to have more annotations which can impact the tuning process. The results may be different if a future study attempts to rebalance the data set based on annotations by rappers. Another reason why performance might be impacted when considering specifically "curated" annotations is that these annotations tend to have more url links as well as references to other artists and songs. These links and references take up additional tokens during the fine tuning process while adding noise to the actual meaning. Comprehensive

[1] https://www.theguardian.com/music/2015/jul/22/ghostface-killah-action-bronson-beef-video-teddy-pendergrass

Prompt	['annotate: Beef rap, could lead to getting teeth capped']

Model	**Generation Text**
T5	['Beef rap is a slang term for a slang term for']
Pegasus	['“Beef rap” is a slang term used to describe a beef between two or more people. In this case, “teeth caps” are used to protect the teeth of the person who is beefing with him or her.']
BART	['Beef rap is a type of rap where rappers talk **** about each other behind each other’s back and use violence as a way to express their anger and frustration. In this case, Action Bronson is saying that he raps about beefing with other rappers so much']

Table 2: The prompt is very simple, I only add the prefix "annotate :". Despite receiving the same prompt, the text generated is very different. See more examples in the Appendix

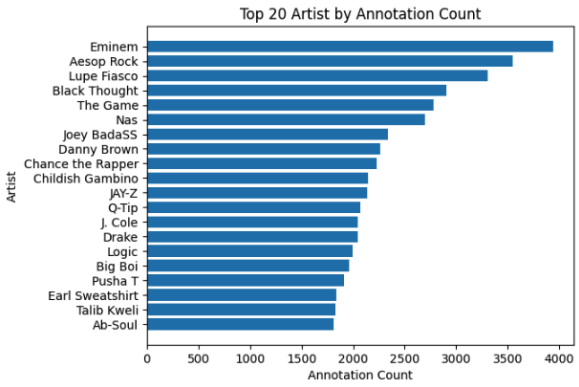

Figure 2: The top 20 rappers account for 40k annotations which can skew the data. Some rappers prefer certain topics and have larger bodies of work than others

annotations such as these tend to garner more votes on the genius platform but can potentially obscure the meaning of the annotation.

Considering the performance of the curated 80k dataset against the full dataset of 160k records are not conclusive across all 3 models, T5 and pegasus only see marginal changes in performance with increases training set sizes where BART sees a relatively significant increase

Model	non-curated (16k)	curated (16k)
Bart	0.176	0.173
Pegasus	0.167	0.167
T5	0.146	0.153

Table 3: Curating more highly rated annotations data does not seem to improve results.

4.3 Results: Training Parameter Tuning

Due to colab resource constraints, the scope of this task was limited to BART because it was the highest performing model. After increasing the tuning max_length from 60 to 128, the model saw additional performance improvement (see chart). This stands to reason as this increased the amount of text that model can train on. Interestingly, the performance with 16k records trained with 128 max_length was comparable to performance with 160k records trained at 60 max_length. Considering the computational trade off between additional records and token size would be an interesting topic for future study.

Model	max_length60	max_length128
Bart 16k	0.176	0.181
Bart 80k	0.174	0.187
Bart 160k	0.183	0.196

Table 4: Curating more highly rated annotations data does not seem to improve results.

4.4 Problem Observations

Some problems that I noticed that were out of scope of the project included: text degeneration, hallucination, and over reliance on certain phrases. In the case of text degeneration, the models may generate the same text over and over again rather than continuing the sentence. In the case of hallucination, the models may falsely assume that a certain rapper is the writer of a lyric or produce a false website link. Lastly, the models will often assuming that one song

or a series of words is a reference to another.

5 Conclusion

The study was able to solve the problem of identifying an LLM and fine tuning it for seq2seq annotation of rap lyrics. Using an ensemble of evaluation metrics including bleu, rouge f1, meteor, and cosine similarity, BART demonstrated superiority to both T5 and Pegasus. This was manifested in not only better annotations, but also faster convergence and higher performance even with less data used for fine tuning. The study also showed that in the absence of additional training rows, additional context during the finetuning process can help improve performance.

6 Future Work

1. Applying prompt tuning would likely result in better performance. For example, including the song title, artist, and additional context from the source song would probably improve results.

2. Given that annotations that are rated more highly tend to be longer, I would consider adding additional tokens at training and revisiting whether or not curated annotations can outperform less highly rated annotations.

3. Obtaining lyrics and annotations from additional rappers could improve the underlying data.

4. Taking the best BART model and training that model on poetry annotations could yield better analysis of metaphors, similes, and other wordplay.

References

[1] Brent Ju Andrew Li Wesley Tjangnaka. *Semantic Understanding of Genius Music Annotations*. 2023.

[2] Michael Fell et al. "Song Lyrics Summarization Inspired by Audio Thumbnailing". In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. Ed. by Ruslan Mitkov and Galia Angelova. Varna, Bulgaria: INCOMA Ltd., Sept. 2019, pp. 328–337. DOI: 10.26615/978-954-452-056-4_038. URL: https://aclanthology.org/R19-1038.

[3] Nikola I. Nikolov et al. "Rapformer: Conditional Rap Lyrics Generation with Denoising Autoencoders". In: *Proceedings of the 13th International Conference on Natural Language Generation*. Ed. by Brian Davis et al. Dublin, Ireland: Association for Computational Linguistics, Dec. 2020, pp. 360–373. DOI: 10.18653/v1/2020.inlg-1.42. URL: https://aclanthology.org/2020.inlg-1.42.

[4] Lucas Sterckx et al. "Break it Down for Me: A Study in Automated Lyric Annotation". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2074–2080. DOI: 10.18653/v1/D17-1220. URL: https://aclanthology.org/D17-1220.

[5] Mor Ventura and Michael Toker. *TR-BLLmaker – Transformer Reads Between Lyrics Lines maker*. 2022. arXiv: 2212.04917 [cs.CL].

A Appendix

Model Name	Name	Datasize	tuning max_length	num beams	max length	no_repeat ngram_size	Rouge	Bleu	Cosine Sim	Meteor	Ensemble score
T5	mini	16k	60	4	100	4	0.159	0.020	0.278	0.129	0.146
T5	mini_curated	16k	60	4	100	4	0.163	0.023	0.284	0.144	0.153
T5	curated	80k	60	4	100	4	0.165	0.023	0.289	0.145	0.155
T5	full	160k	60	4	100	4	0.163	0.021	0.294	0.134	0.153
Pegasus	mini	16k	60	4	100	4	0.173	0.026	0.316	0.153	0.167
Pegasus	mini_curated	16k	60	4	100	4	0.172	0.026	0.314	0.154	0.166
Pegasus	curated	80k	60	4	100	4	0.173	0.026	0.316	0.157	0.168
Pegasus	full	160k	60	4	100	4	0.174	0.027	0.318	0.156	0.169
Pegasus large	mini	16k	60	4	100	4	0.159	0.020	0.299	0.129	0.152
Pegasus large	curated	80k	60	4	100	4	0.167	0.024	0.309	0.145	0.161
Pegasus xsum	mini	16k	60	4	100	4	0.153	0.018	0.290	0.121	0.146
BART	mini	16k	60	4	100	4	0.179	0.028	0.330	0.164	0.176
BART	mini_curated	16k	60	4	100	4	0.177	0.028	0.327	0.161	0.173
BART	curated	80k	60	4	100	4	0.179	0.028	0.323	0.167	0.174
BART	full	160k	60	4	100	4	0.189	0.031	0.338	0.174	0.183
BART	mini	16k	128	4	100	4	0.180	0.029	0.341	0.175	0.181
BART	curated	80k	128	4	100	4	0.185	0.031	0.342	0.191	0.187
BART	full	160k	128	4	100	4	0.197	0.034	0.355	0.196	0.196

Figure 3: Full Tables of experiments and results