Semantic Understanding of Genius Music Annotations

Stanford CS224N Custom Project

Andrew Li
Department of Computer Science
Stanford University
andrewvli@stanford.edu

Wesley Tjangnaka
Department of Computer Science
Stanford University
wesleytj@stanford.edu

Brent Ju
Department of Computer Science
Stanford University
brentju@stanford.edu

Abstract

In our project, our goal is to design a Seq2Seq model to handle the task of generating annotations for song lyrics, a creative natural language task proposed by Ventura and Toker. To achieve this goal, we modify the original author's approach to use an autoregressive GPT-2 decoder, electing to use a Seq2Seq T5 model instead while providing the model additional context with our addition of a named-entity recognition model and an information retrieval system. Our two best models achieve Rouge-1 scores of 0.657 and 0.566 and cosine similarities of 0.163 and 0.236 to our test annotation dataset, outperforming the previous TRBLLmaker model that attempts the same task of lyrical analysis generation.

1 Key Information to include

- Mentor: Heidi Zhang
- External Collaborators: None
- Sharing project: No

2 Introduction

2.1 Motivation.

The field of natural language processing (NLP) has made significant advancements with the development of Transformer models, which have been proven to handle sequence-to-sequence tasks and long-range dependencies effectively. Researchers have deployed transformers in numerous generative tasks such as translation, summarization, dialog, and question answering, with a focus on generating output that can be found in the input. However, we see persistent challenges remaining in the task of abstractive summarization — a process concerning the interpretation, analysis, and compression of text into shorter summaries that emphasize the most important points — which is heavily dependent on solving complex problems involving semantic representations and contextual meaning. Moreover, these output-input methods cannot effectively handle complex texts, hidden clues, or subtle implications, leading to poor performance when tasked with "reading between the lines" and understanding the environment, context, and semantics of textual art such as poems and song lyrics.

In this project, we pursue improvements to existing attempts at the task of generating annotations for song lyrics; we specifically propose a series of modifications to the TRBLLmaker model presented in Ventura and Toker (2022). The original work done presented a generative model utilizing a

decoder-only transformer GPT-2 and compared its results to that of the encoder-decoder architecture of the T5 model. However, the performance of their model was limited by incorrect identification of relationships between subjects of the lyrics and between artists that wrote the song, as well as by its tendency to produce false information related to the artist and the cultural environment pertaining to the song's release. Our original contribution offers a novel usage of named entity recognition paired with an information retrieval system to provide greater context for input song lyrics.

3 Related Work

Existing work has been conducted on the construction and annotation of songs and song lyrics corpora, as well as their usage for various downstream tasks. For instance, Rodrigues et. al present a web-scraped English song lyrics corpus which they propose for usage in automatic generation of lyrics and poems (Rodrigues et al., 2019). Fell et. al offer an elaboration on this goal, presenting a corpus of song lyrics enriched with metadata extracted from web music databases and with extractions on relevant information from song lyrics, such as their structure segmentation, topics, explicitness, and conveyed emotions (Fell et al., 2019). Numerous subsequent studies have focused on the utilization of such corpora for tasks ranging from the classification of explicit song lyrics (Rospocher, 2022; Rospocher and Eksir, 2023) to semantic analysis and the detection of mood or emotion (Donnelly and Beery, 2022; Naseri et al., 2022).

However, comparatively limited research has been done into the specific task of generating annotations from song lyrics, which requires a model to "read between the lines" of a song and understand the semantics, environment, and context of the text. We examined several prior attempts at this task. Sterckx et. al compared the performance of standard SMT models with Seq2Seq models in automated lyric annotation, finding that Seq2Seq models demonstrated greater potential in generating fluent and informative text that went beyond the lyrical content; we drew from their suggestions to inject further structured and unstructured external knowledge as context for automated annotation through our model's proposal of named entity recognition paired with information retrieval (Sterckx et al., 2017). However, the prior research we use as a benchmark for our work is "TRBLLmaker: – Transformer Reads Between Lyrics Lines maker" by Mor Ventura and Michael Toker (Ventura and Toker, 2022).

The original authors elected to finetune a decoder-only GPT2 model on their own custom dataset containing around 60,000 samples from the music website Genius.com. In addition to providing a dataset, the paper also makes a novel attempt at a new task of generating meanings of songs. The authors emphasize that while significant research has been spent on sentiment analysis of lyrics or usage of text-to-text generation to complete songs in a certain fashion, there exists very little work in using natural language processing to decipher the implicit meanings behind songs, a task that poses difficulties even for humans. Ventura and Toker experiment with various strategies of prompting the model in order to achieve optimum outputs, and they find that fine-tuning their model by posing the task of generation of annotation as a text completion problem with metadata on the song achieves the highest results. This strategy resulted in achieving a Rouge-1 score of 0.47 and a cosine similarity to the annotations of 0.17. Various decoding strategies such as beam-search, greedy search, and top-K sampling were employed and compared, but the results were not significantly variable among the different trials. The authors also proposed a custom metric for evaluating the model-generated annotations that punish memorization or excessive generalization to the lyrics, which will be covered further in the section on evaluation methods.

4 Approach

4.1 Tasks

Our task for model development and evaluation is text-to-text generation. Given an input of song lyrics, our goal is to generate an annotation for the specific lyrics that explain the meaning while also giving light to potential subtextual information in the song. Possible subtextual information could include word play, entity linking to real-world references, and more.

4.2 Methods

We followed the training procedure of the original paper in fine-tuning a GPT-2 decoder to our task as an initial baseline for comparison. Later onwards, we chose to use and finetune the encoder-decoder structure of a T5 model over the autoregressive GPT-2 model as supported by Sterckx et al. We also hypothesized that song lyrics would often be more informal with implicit meanings while the annotations were more analytical and objective in content, and so a Seq2Seq model would be more fitting than a decoder-only model.

Our model also introduces the novel additions of a named entity recognition and information retrieval system. The original literature faced difficulties with the model producing annotations that were misidentifying relationships between objects, namely musical artists, in the generated annotations. In our efforts to improve upon this issue, we use Stanford NLP's Stanza NER pipeline Qi et al. (2020) to identify named entities. These entities will then be passed as entries to Wikipedia's API to retrieve pages. The API will always either return an exact match, or a collection of relevant matches that were close to the original input. The returned pages from the API will then be fed into a BERT model for sentence ranking; either the most relevant sentences in a perfectly matched Wikipedia page or a group of relevant sentences from a collection of near matches will be collected in order to provide additional context for the model to process and use for generation.

Finally, this is fed into the Open-AI's GPT-3 API. We prompt it with examples on how to merge the two prompts, and it seems to be more successful with few-shot learning.

4.3 Baselines

For our baselines, we opted to fine tune a pretrained version of GPT2 from HuggingFace on our lyrics dataset. We had originally planned to use a custom encoder-decoder structure using RoBERTa, but, because RoBERTa was not well suited for text generation and often gave us empty results and generations, we opted for GPT-2 as the stronger model of choice to set a baseline. The finetuned GPT-2 model achieved a ROUGE-1 score of 0.12927 and a BLEU score of 0.009.

4.4 Model design and architecture

The major component of our model is the Seq2Seq T5 model for generating the annotation itself. Our lyric inputs are first tokenized using the HuggingFace T5-small tokenizer to a length of 256 tokens due to memory constraints on our GPU instances.

In a separate branch, the input is fed through a fine-tuned Stanza for named-entity recognition in order to extract relevant entities, if any, in the lyrics. These entities will then be passed into a request to the Wikipedia API to retrieve relevant pages to perform sentence ranking upon for additional context to be used in our model.

Once the T5 annotation is generated and the sentence ranking is completed, we start prompt engineering GPT-3 using the OpenAI API. We ask the model to treat the sentences from the Wikipedia API as ground truth, and include portions of those sentences as appropriate into our T5 generated output, which we then use as our final output annotation. See Figure 1 for a more detailed graphic of the model architecture.

5 Experiments

5.1 Data

To generate annotations for an associated lyric of a song, we used the Cornell University Genius Expertise dataset, provided by Austin Benson (Lim and Benson, 2021). The dataset contains over 400,000 entries containing various song lyrics and associated annotations from the Genius Music online community. We processed the data by first isolating the song lyrics as the input to our model. Because a set of lyrics can have multiple annotations associated to it from various different users, we leveraged the fact that users can vote upon which annotations they

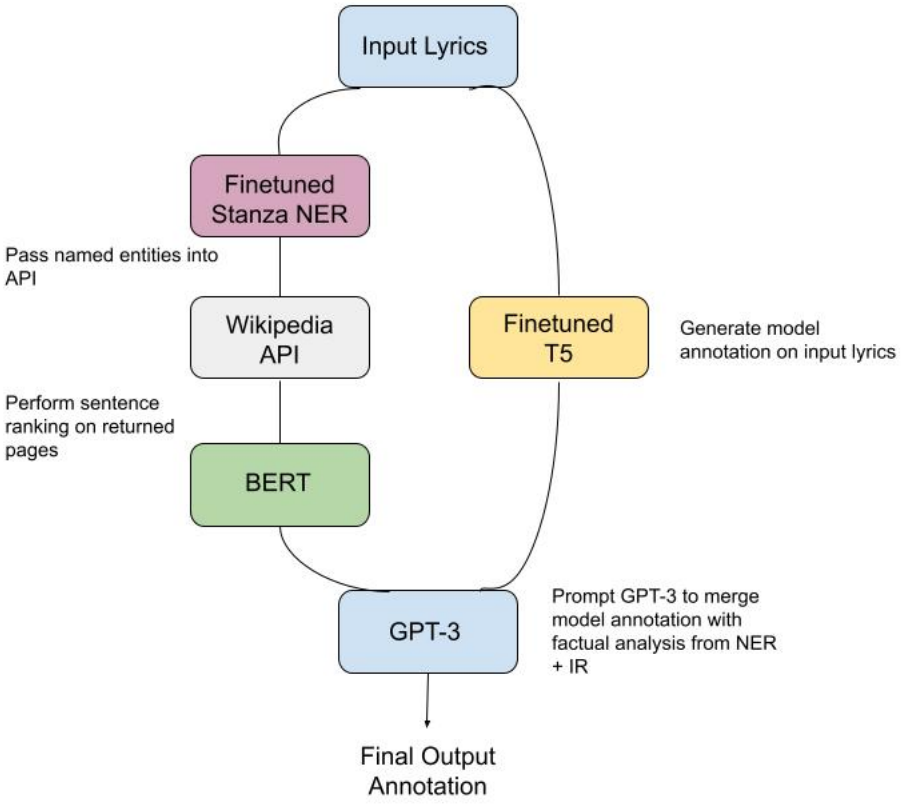

Figure 1: Workflow of our model.

favor to select only the top annotation for each lyric as our target annotation for the model at train time.

To fine-tune the Stanza NER model, we use the Entity Linking for the Music Domain (ELMD) dataset, which provides annotated artist biographies, song names, and artist names (Oramas et al., 2016). The purpose of using this dataset to fine-tune the NER model is to provide better recognition of song/artist/band names, and possibly allow better performance in a music specific lens. We processed the data by converting it into a format that would be usable by Stanza for training - that is, we tokenized each JSON sentence and attached a NER label to each token of the sentence. We also split this into a train, test, and dev set to be used for training.

5.2 Evaluation method

In order to penalize the model from generalizing too much to the lyrics and spitting out meaningless repetitions of the original input, the authors created a custom metric that weights Rouge scores and cosine similarities of the model's output to the original annotations as well as the original lyrics:

$$\textbf{Total Score} = \sum \alpha_i \cdot m_i$$

$$= \alpha_1 \cdot \textbf{Rouge} + \alpha_2 \cdot \textbf{cos}_{\textbf{model output, annotation}} - \alpha_3 \cdot \textbf{cos}_{\textbf{model output, lyrics}}$$

For consistency purposes, we followed the original paper's choice to set the weight to 0.5.

In addition to this custom metric, we elected to calculate the raw Rouge-1 and cosine similarity scores to compare against the original paper.

5.3 Experimental details

For finetuning the Stanza NER model, we set a maximum number of gradient descent steps to 200,000. The hyperparameters we chose were an initial learning rate of 0.1, hidden layer size of 256, word and character embeddings size of 100, max gradient norm of 5 for gradient clipping, a batch size of 32, and dropout rate of $p = 0.5$. We had an LR-decay rate of 0.5, minimum learning rate of 10^{-4}, and patience of 3.

Due to computational and memory constraints, we elected to use the pretrained T5-small model from HuggingFace with 60 million parameters.

We first concatenated "Summarize: " to the beginning of each of our input lyric strings to establish the task at hand for the T5 model. Both the input and target annotations were then passed into the T5-small tokenizer at a maximum length of 256 tokens due to GPU memory limitations.

Our model was finetuned on 10 epochs which ran for $\sim$ 7.5 hours on a single Amazon G5 instance. We used cross-entropy loss as our objective function and we used the Adam Optimizer with an initial learning rate of 0.0001.

For text generation, we used the provided generate function included with the T5 model API. We performed beam-search decoding with 6 beams and bounded the range of outputs to be between 24 tokens and 512 tokens. Our choice of having a lower bound on annotation length was to discourage the model from making short observations that didn't encapsulate more information about the lyrics: for example, when given the lyrics, "My Louboutins new, so my bottoms they [are] redder," the model will simply generate the blurb that "Louboutin is a luxury shoe brand."

5.4 Results

We are happy to report that our model outperformed the prior literature by a fairly significant margin. While performing calculations and comparisons against Ventura and Toker's results, we actually noticed inconsistencies in how their custom total score metric was actually calculated. Their results are included below; however, the main point of concern is that their Rouge F1 and cosine similarity scores are unable to produce a sum that equals their reported total scores.

The TRBLLMaker paper used 2 prompting strategies to generate text, both of which performed best under beam search decoding. The first strategy is what the authors refer to as lyrics meaning prompting, where the inputs to the GPT model were of the form: "lyrics: [input] meaning: [annotation]." The other strategy used was question-context prompting of the form "question: what is the meaning of [artist]'s song [title]? context: [lyrics]. answer: [annotation]."

Model	Decode Strategy	Rouge F1	Cosine Similarity (Annotation)	Total Score
T5 + NER (finetuned) + IR	Summarize + merge	0.566	0.236	0.293
T5 + NER + IR	Summarize + merge	0.6567	0.163	0.32*
T5 + finetuning	Summarize	0.167	0.1246	0.08*
TRBLLMaker	Lyrics meaning prompt	0.038	0.21	0.55[1]
TRBLLMaker	Question context prompt	0.042	0.23	0.65[1]
GPT-2 (finetuned)	Question/Answer prompt	0.129	N/A	N/A

*If a model generalizes too strongly to the input lyrics, the custom total score is capped to 0.5 ·Rouge F1 score.

[1]Questionable calculations from the original TRBLLmaker paper.

We believe the improvement in results is attributed to the inclusion of additional context via our named-entity recognition and information retreival system. This addition simultaneously reduces the number of erroneous facts in the generated annotation while also providing additional information that likely contributes towards token matches that benefit the Rouge-1 and cosine similarity scores.

6 Analysis

6.1 General Observations.

We have noticed that the model performs exceptionally well when the lyrics involve luxury or designer objects. For example, when asked to give an annotation for the following lyrics for Lil Baby's "Yes Indeed:"

"Cartier glasses I won't even peek at you, yellow Ferrari like Pikachu,"

The model correctly identifies that Cartier is a luxury jewelry brand and also that the lyrics make a "reference to the famous Ferrari, which is known for its yellow and red colors." What's interesting is that despite the fact that the lyrics explicitly state that the Ferrari is yellow, the model appears to have gained the knowledge from pretraining that red is a flagship color that is associated with the Ferrari brand.

We have also found that the model tends to generalize and overfit a pattern of assuming that a song lyric is a reference to another artist, song, or real world event. 12% of the model-generated annotations contain the phrase "This is a reference to," which precedes some prediction about what the lyrics are in reference to.

In a similar vein, our dataset appears to have been composed with a majority of rap music lyrics, specifically of songs by Jay-Z or referencing the Brooklyn rapper. The model often will incorrectly predict that abstract entities inside of a song are referring to Jay-Z or his alias "Hov" approximately 25% of the time.

Unfortunately, due to the nature of some of the lyrical contents that the model was trained upon, we will often find annotations containing violent content as well as misogynistic/sexual commentary. With more time, we would like to process and verify the content of the lyrics and annotations to hopefully reduce the amount of harmful content that the model gets trained on.

Something else that was interesting is that the finetuned NER model seemed to perform worse than the pretrained NER model. After examining the data more, we noticed that although the finetuning dataset provided further insight into named entities in the musical domain, it missed out on annotating other named entities such as names (in the non-musical context). For instance, while qualitatively analyzing the outputs from our finetuned versus regular NER models, we noticed that while it correctly identified names of bands or artists that were uncommon, it would more frequently misidentify cities and locations, since they weren't as correctly labeled in the finetune dataset.

7 Conclusion

7.1 Summary.

Our improved results upon prior literature highlight the potential of transformers and large-language models in natural language tasks such as abstract and artistic summarization. Our approach, a Seq2Seq model that provides additional context for automated annotation generation through named-entity recognition and information retrieval, underscores the potential of external information injection in improving the performance of state-of-the-art models. While there is still much work to be done, with greater magnitudes of data being required, before a deep model can discover implicit meanings in song lyrics at the level of human analysis, it is undoubtedly exciting to see the possibilities available given the amount of training and work put into our project. The further implications of this research are fruitful, with use cases stretching beyond artistic interpretation and offering the potential for further improvement in handling textual variety in more traditional NLP tasks (Sterckx et al., 2017).

7.2 Limitations.

The major limitation of this project was due to the compute power available to us in the training of our model. While we are extremely grateful for the provided credits and could not have completed

this project without the generous donation of AWS credits, we realize that more power would have allowed us to fine-tune larger versions of our models that may have achieved higher performance.

Because the dataset we used was not properly cleaned and contained a lot of HTML tags, images, and website URLs, the model would often incorrectly try to predict when to insert such elements and would often also create links to non-existent sites. Future work could include processing the target annotations further to prevent this issue from arising and affecting our model at train time; however, thorough cleaning of the data was not feasible given the deadlines and scope of this project.

Furthermore, we could have improved upon the information retrieval within our project - the Wikipedia API takes really specific phrases to output the exact document of interest. Though we attempted to adjust for this by also querying the possible disambiguations and then ranking those, and choosing the most similar one, often times the Wikipedia API couldn't find a match for an identified named entity. For instance, the named entity recognition model correctly identifies the city "Compton" from the lyrics of Kendrick Lamar and Dr. Dre's song "Compton." However, the precise Wikipedia entry to query its corresponding page is "Compton, California." Even though our model accounts for these unexact matches, some still aren't picked up on when searching through the possible disambiguation results. As a result, this added context would be missed out on when forming the final summarization.

We were also limited by our calls to the OpenAI API, as we weren't able to run the entire evaluation set on one run. We attempted a temporary solution by writing our generations onto a document, and running the evaluation script on the document after making all of our generations from different computers and different API keys. However, we weren't able to test the entire evaluation set, though we did manage to test a very large subset of it.

References

Patrick Donnelly and Aidan Beery. 2022. Evaluating large-language models for dimensional music emotion prediction from social media discourse. In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 242–250.

Michael Fell, Elena Cabrio, Elmahdi Korfed, Michel Buffa, and Fabien Gandon. 2019. Love me, love me, say (and write!) that you love me: Enriching the wasabi song corpus with lyrics annotations.

Derek Lim and Austin R Benson. 2021. Expertise and Dynamics within Crowdsourced Musical Knowledge Curation: A Case Study of the Genius Platform. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 373–384.

Shahrzad Naseri, Sravana Reddy, Joana Correia, Jussi Karlgren, and Rosie Jones. 2022. The contribution of lyrics and acoustics to collaborative understanding of mood. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 687–698.

S. Oramas, Anke L. Espinosa, M. Sordo, H. Saggion, and X Serra. 2016. Elmd: An automatically generated entity linking gold standard dataset in the music domain. In *LREC 2022*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Matheus Augusto Gonzaga Rodrigues, Alcione de Paiva Oliveira, and Alexandra Moreira. 2019. Development of a song lyric corpus for the english language. In *Natural Language Processing and Information Systems: 24th International Conference on Applications of Natural Language to Information Systems, NLDB 2019, Salford, UK, June 26–28, 2019, Proceedings 24*, pages 376–383. Springer.

Marco Rospocher. 2022. On exploiting transformers for detecting explicit song lyrics. *Entertainment Computing*, 43:100508.

Marco Rospocher and Samaneh Eksir. 2023. Assessing fine-grained explicitness of song lyrics. *Information*, 14(3):159.

Lucas Sterckx, Jason Naradowsky, Bill Byrne, Thomas Demeester, and Chris Develder. 2017. Break it down for me: A study in automated lyric annotation. *arXiv preprint arXiv:1708.03492*.

Mor Ventura and Michael Toker. 2022. Trbllmaker – transformer reads between lyrics lines maker.