# Intro:

Good [morning/afternoon/evening], ladies and gentlemen,

In an era where songs can effortlessly get stuck in our heads and where “waxing lyrical" means to describe with great enthusiasm and eloquence, understanding the impact of song lyrics has never been more significant. It's not just about the beat and melody; it's the words that give music its soul, enabling us to connect, emote, and find meaning in the notes.

Music is often regarded as a universal language. It transcends geographical borders and cultural differences, connecting people across the globe through its emotional and tonal qualities. Now, think of a song that has moved you, perhaps to tears, or to the dancefloor. It's not just the melody; it's the lyrics that create this powerful connection. They have the astonishing ability to tap into our emotions in a way that resonates deeply with us.

While the lyrics of songs may not always be universally understood, the emotions and stories they convey have the power to touch the hearts of individuals around the world. Whether you're in the heart of New York City or the serene Japanese countryside, listening to the strings of a guitar or the rhythmic beating of a drum, the power of lyrical verses unites us all. A song can touch your soul even when you don't understand the words. It's this universal appeal that makes music one of the most potent forms of art.

Today, I invite you to embark on a journey into the captivating world of music and lyrics, where the power of today’s cutting-edge technology converges with the centuries old art of storytelling. We'll explore a groundbreaking topic, "Developing a Machine Learning Model for Automatic Song Lyric Analysis," and uncover the remarkable ways in which this fusion of art and science will revolutionize the music industry and our understanding of song lyrics.

Consider this: songs are, at their core, stories set to music. Lyrics unfold narratives, transporting us to different times, places, and experiences. They narrate tales of love and loss, hope and despair. Just as a great book or film can captivate us, songs like "Imagine" by John Lennon or "Bohemian Rhapsody" by Queen are epic stories in themselves.

They are more than just words; they are reflections of the society and culture from which they were born. From the protest anthems of the '60s to the socially conscious rap of today, lyrics have been a potent force for change. They shed light on societal issues, advocate for justice, and express the ideologies of their time.

The goal is to develop a model that is capable of providing a comprehensive and linguistically fluent understanding of the lyrics of a given song. It can be applied to various scenarios, such as:

1. Lyric-based song recommendations, for example: If a user listens to songs having a positive and upbeat message, more such songs can be recommended by the model.
2. Parental controls: Parents and guardians can prevent songs with adult undertones and themes from reaching their children’s ears.
3. Education: The model can be used in a learning environment by teachers and students alike to help understand poetry.
4. Song writing: The application does not have to be limited to just the listeners of a song, as artists can use interpretations to help in their creative processes or gain inspiration.

# Why is it challenging?

Our topic has the potential to be one of the hardest, most intricate challenges within Natural Language Processing. It goes beyond text comprehension and question answering, where responses can often be extracted directly from the text. Attempting to capture true essence of a song extends far beyond its lyrics; it involves comprehending the emotions, context and nuances that were poured into the song. The following challenges are a direct consequence of songs as an art form, present for both human listeners and our machine learning models.

[next]

## Subjectivity

All forms of art are inherently subjective. What resonates with one person might not have the same impact on another.

[next]

I believe it was Huey Lewis and the news who sang “*It’s a curious thing, what makes one man weep, will make another man sing*.”

Therefore, describing an interpretation “correct” is simply not possible. But, beneath this subjectivity, there exist common themes and emotions that songs often explore. These provide a bridge between individual interpretations and can guide listeners towards a more collective understanding of their messages.

[next]

For machine learning models, deciphering these diversities is a crucial part of song interpretation. These models must recognize the recurring motifs and concepts that serve to connect songs across genres and cultures.

[next]

## Figurative Language

One of the primary challenges for human listeners is navigating through the use of figurative language. Songs often rely on metaphors, symbolism, irony, satire, and poetic language. Deciphering when an artist employs these tools adds complexity to understanding. What might seem straightforward could carry layers of meaning, resulting in various interpretations.

[next]

Consider a verse from I ran so far away (an 80s song made by A Flock Seagulls – yes, that’s their name), *“A cloud appears above your head. A beam of light comes shining down on you.”* This may seem like a romantic description, but believe it or not, the line, song and entire album deal with UFO abductions. Thus, lyrics are not always to be taken at face value.

[next]

Machine learning models, too, encounter these hurdles when interpreting songs. They must navigate the creative and often abstract nature of songwriting, which can be especially challenging for algorithms that were not trained to grasp the subtleties of human expression.

[next]

## Artistic Intent

We also grapple with the fact that artists themselves are sometimes intentionally cryptic, with examples ranging from baking in wordplay to the reason for writing the song in the first place.

[next]

Consider Kurt Cobain's conflicted relationship with "Smells Like Teen Spirit" – the single that rocketed them to 90s stardom. While it may sound like an anthem of rebellion, Cobain's original intent was quite the contrary. He wanted to create the ultimate parody pop songs, mimicking generic elements, such as repetitive choruses, that prioritize catchiness over substance.

This revelation underscores how artists have the remarkable ability to craft cherished melodies, often layered with profound meanings, even if such depths weren't their primary aim. But this complexity can make it challenging for listeners and models to trust that they've unravelled the true essence of a song.

[next]

These challenges set the stage for why we look to machine learning as a powerful tool for song interpretation. By harnessing the capabilities of these algorithms, we can explore new horizons in understanding the intricate nature of music and lyrics – and perhaps surpass human based results as models have done in other NLP topics.

[next]

# How do we go about solving this? Lit review

So how do we go about solving this problem? Let us look at what those who came before have accomplished.

[next]

## Attention

Attention mechanisms gained significant attention with the introduction of the 2014 paper "Sequence to Sequence Learning with Neural Networks" . They introduced a novel attention mechanism that allowed the model to shift its focus on different parts of the input sequence while generating an output sequence.

They are designed to mimic exactly how attention works in humans. It allows models to focus on the most relevant parts of the input data while downplaying less important information. For instance, when our model receives a song, the mechanism may assign less weight to the chorus, making the model focus more on the verses to construct an interpretation instead.

[next]

## Transformers

The 2017 paper "Attention Is All You Need" presented the Transformer architecture. Its attention mechanism was pivotal in its success and laid the foundation for modern deep learning models. A transformer utilizes 2 kinds of attention:

[next]

1. Self-attention refers to calculating a vector representation of each word, in turn making use of 3 other vectors known as query, key and value. Consider how a sample sentence can be broken down into such a vector representation. ntuitively, q represents a question, e.g.: Who is running for president? Then, q(one) by k(one) = “Trump” would be the answer. The highest value of q \* k represents the best answer and would be assigned to the “v” vector.

[next]

1. (2) Multi-head attention simply means that we calculate attention for the same sequence multiple times. Each time, the calculation remains the same, but we use different weights. These can correspond to asking different questions, for example, “What is happening?” or “When di this event happen?” Hence, we have attention A one, A two and so on for each set of weight matrices.

[next]

But what does all this math get us? Well, it means that transformers can:

* can consider the entire input sequence when generating the output. Traditional recurrent neural networks (RNNs) process sequences sequentially and are limited by the context window, while Transformers can capture relationships between distant words or tokens.
* utilize multi-head attention to focus on different aspects of the input data. Each "head" can attend to different parts of the sequence, enabling the model to learn multiple types of relationships within the data.
* less susceptible to the vanishing gradient problem when dealing with very long sequences, which is a common challenge in deep learning.
* And most importantly, they have achieved state-of-the-art results in a wide range of NLP tasks, including machine translation, text summarization, sentiment analysis, and question-answering.

Now we go on to introduce 2 specific implementations of the transformer architecture: BERT and BART

[next]

## BERT

A large shortcoming of the base transformer architecture is that the encoder is only capable of reading tokens from left to right. Language is complicated and dynamic, having many instances where even the whole sentence is required to understand the context of a single word.

Therefore, being able to read tokens in only a single direction hampers model’s ability to understand context. The Bidirectional Encoder Representations from Transformers or BERT is a model made for NLP applications using largely the same architecture as the base Transformer model.

[next]

The pre-training framework involves gigantic text corpuses (namely English Wikipedia and the BooksCorpus - amounting to about 3.3 billion words) being applied to two tasks in equal measure: Masked Language Modelling and Next Sentence Prediction (NSP). The former concept is how BERT able to utilize bidirectional learning: It predicts a masked/hidden word (that the model sees as a “MASK” token) using the words to its left and right. The latter is how BERT learns the relationship between sentences: It must predict if a target sentence is semantically likely to follow a previous sentence. It’s pre-training process [4] BERT uses a special classification and separation tokens, which denote the start of each of the first and second sentence in the pair respectively.

[next]

The evaluation methods consisted of using the:

* Stanford Question Answering Dataset / SQuAD challenge wherein we predict where the answer to a question lies in a passage from a Wikipedia page) measured by EM (exact match) and f1- scores.
* Situations With Adversarial Generations / SWAG has the model choose the most likely event given a sentence out of 4 possible options).
* And the General Language Understanding Evaluation / GLUE benchmark which is used to evaluate language models on a range of tests measured by metrics prevalent to each task.
* In all cases, BERT outperformed the previous models and human performances (where applicable).

This is a powerful model for NLP applications and thus well suited be used for our lyric interpretation task.

[next]

## BART

BART uses a base transformer architecture with the only change being that they use GeLU as the activation function. It is also similar to BERT in that it incorporates bidirectional learning but differs in 2 ways: Cross attention (which we will come to later in an easier example) is done at the last encoder hidden layer. And no additional feed-forward neural network before any word prediction.

[next]

Pre-training of the BART model was done using text from Wikipedia as well as certain select books It has 2 stages:

The text is first transformed using various methods with each transformation attempting to “teach” the model a different way of understanding context. These include:

* Token masking: Similar to BERT, tokens are masked/hidden from the model to allow for the model to understand the context of words within a sentence.
* Token deletion: As opposed to masking, the word is removed completely without adding a MASK token. The purpose is to get the model to predict which words are missing from which positions.
* Sentence permutation: The sentences within a document are shifted around to appear in a random order. Thus, the model is allowed to learn the context sentences based on other sentences.
* Document rotation: A document is shifted such that it begins with a randomly selected token. This teaches the model to predict what the document should start with.
* And lastly, we have Text infilling: An amount of the text of a document is selected and replaced with a single MASK token. The purpose is for the model to learn how many tokens have been replaced.

The goal of all these transformations is for the model learns to undo these transformations in order to rebuild the original text. Hence why BART is referred to as a “de-noising” model.

[next]

BART has been evaluated on several tasks:

* Outperforms BERT on the SQuAD challenge and several other GLUE tasks.
* On generative tasks such as summarization, dialogue (where the model generates responses based on a given personality) and Abstractive Question Answering, BART outperforms the previous best works.
* On translation tasks, BART was shown to be prone to overfitting, but still performed above the baseline.

BART is a PTM that is especially suited for NLP tasks because of the methods used in pre-training. It is well-adapted to comprehension tasks and text generation – which are critical elements of lyric interpretation. We can make use of the BART “large” model fine-tuned on the CNN-Daily-Mail dataset and has more parameters than the base version.

[next]

## PEGASUS

The PEGASUS model is also built on the base transformer architecture, but it aims to improve on the BART model in the realm of pre-training and masking. Its main usage is meant for abstractive text summarization - which is where generated summaries are able to incorporate new words learned from the large corpus they are trained on. Its pre-training is uses BERT’s Masked Language Modelling and a new method known as Gap Sentences Generation (GSP).

Unlike BART, PEGASUS masks entire sentences as opposed to a masking a sequence of words within a single sentence. The intuition behind this approach is that a summarizer model pretrained as an extractive type (where a model does not generate new words in its summaries) would merely only be able to copy sentences and not display a full understanding of the context.

[next]

The GSP process involves selecting the most important sentence of a document and replacing it with a single “MASK1” token. The other sentences also contain some random masking of one “MASK2” token per word as per a Masked LM approach. These techniques are applied simultaneously – similarly to how BERT uses Masked LM and NSP in conjunction.

Pre-training of the PEGASUS base model utilizes the Colossal and Cleaned version of Common Crawl / C4 (which is a collection of web pages) and HugeNews (a dataset of news articles) that totals to over 4.5 TBs of data.

The PEGASUS large model features more than double the amount of training parameters and further trains on several varied datasets (from the CNN-Daily-Mail to social media posts from the r/TIFU Reddit community). These extra pieces of training data provision for added diversity in terms of length, style of writing, and general abstractive-ness.

[next]

The evaluations consisted of using ROUGE metrics and comparing them to previous state of the art / SOTA abstractive model results. Collectively, the BART models outperformed the previous SOTA. “Collectively” as in 2 PEGASUS large models were used. Each was pre-trained on either the C4 or HugeNews dataset: So the C4 version outperforms some and the HugeNews version outperforms the other previous SOTA results.

Interpreting lyrics can be seen as an abstractive summarization task as we are not trying to summarize the lyrics exactly, but rather get the model to associate the sentences of the comments to verses in the song. We can use the PEGASUS large model, which has has performed well at this task, as a model for our application.

[next]

## BART fusion

To the best of my knowledge, this is the only other known paper that has attempted the task of lyric interpretation. It proposes “BART-fusion”, which uses a linguistic PTM and audio samples from a song. The intuition behind the use of audio is to allow for more information can be derived about the song.

[next]

A second encoder is added to the traditional transformer model to allow for the extraction of an audio representation of the given lyrics. It is able to extract the audio features using a convolutional neural network and also incorporates a multi-head attention sub-layer similar to the base transformer architecture (with a feed forward neural network and normalization sub-layers above it). It differs in that it features multi-head “cross-attention.”

Cross-attention is calculated the same as self-attention, but it can combine two different input sequences (i.e.: the audio and lyrics). It has also been applied to machine translation and image generation tasks, where we need to map a source to target sentence and a prompt to an image respectively.

[next]

For evaluation, their BART-fusion model was compared to and outperformed the baseline BART model on the ROUGE and METEOR metrics and more or less matched it on the BERT-score (which we will also come back to). It has been noted that when the lyrics revolve around difficult subjects that carry complex undertones, both models fail to generate a real, analytical interpretation.

The other important contribution of this paper is their impressive “Song Interpretation Dataset”. It contains the audio and lyrics of over 20,000 songs from a variety of genres and almost half a million human interpretations. It also includes the number of votes for each interpretation. They have preprocessed this dataset to remove short interpretations (which were often found to have nothing to do with the song) and ensure that only quality interpretations are used by taking those having the highest number of votes. For testing purposes, they have created a smaller dataset by removing 800 rows from the original dataset. Note that, unlike the other two known song and interpretation datasets, this has been made available to the public and features much more data (approximately 10,000 and half a million more songs and interpretations respectively).

We can utilize the Song Interpretation Dataset and BART-model approach to song interpretation.

[next]

# Have these achievements come at a price?

"Allow me to borrow a quote from Jeff Goldblum's iconic character, Ian Malcolm, in Jurassic Park: 'Our scientists were so preoccupied with whether or not they could, they didn't stop to think if they should.' This statement resonates not just with scientific endeavours but with any human pursuit, especially when it involves innovation and advancement.

In our quest to push the boundaries of machine learning, it's essential to pause and consider the implications of our work. It’s in moments like these that we turn to different perspectives outside this field for guidance and motivation.

[next]

Consider the Overview effect, which is a fundamental shift in reality, experienced by astronauts upon seeing our planet from space. Many of them have advocated for world leaders to be sent up, because, from the perspective of our daily lives, we hardly think about the Earth really is. Out there, all an astronaut can see is a tiny, helpless blue speck, floating in the endless void, held together by a paper-thin atmosphere. This profound realization emphasizes the importance of understanding and safeguarding our planet.

[next]

How does this relate to our work? Well, the popularity and availably of educational resources for machine learning is growing. It has even got to the point where ML courses are being implemented at a high-school level.

However, this influx of people also means a directly proportional increase in the computing resources required to accommodate a larger community. These resources range from storage to GPUs, which are also being used by research teams to increase the capabilities of models (consider the ever-improving ChatGPT as a prominent example).

The increase in resource usage is unfortunately directly proportional to CO2 emissions and a larger carbon footprint .

[next]

# What can we do about it?

To be more mindful of and address our impact on the environment, we can use the concept of transfer learning. This approach significantly reduces the need for resource-intensive, "from-scratch" training. The usage of Pre-Trained Model’s mean:

* **Fewer Training Epochs**: Pre-trained models, such as BERT and GPT, are already trained on vast datasets, reducing the need for resource-intensive training from scratch. Fine-tuning requires fewer training epochs, which means less energy consumption during model development.
* **Data Efficiency**: Pre-trained models have access to a wealth of information from diverse data sources. Fine-tuning on task-specific data allows for knowledge transfer, reducing the need for large, domain-specific datasets – reducing data acquisition efforts.
* **Reduced Hardware Demand**: By relying on pre-trained models, developers can often use less powerful hardware. This leads to lower resource consumption and a smaller carbon footprint.

Now that we've explored how adopting sustainable practices in machine learning can mitigate our environmental footprint, let's dive into the practical aspect of implementing these principles.

[next]

# Implementation

## Leveraging Pre-Trained Models:

* I've chosen to utilize the Hugging Face platform for accessing pre-trained models, specifically the facebook/bart-large-cnn and google/Pegasus-large. These models serve as invaluable starting points for our project, allowing us to build upon existing knowledge.
* Hugging Face offers a framework for training and deploying models. Its online repository simplifies the storage and retrieval of both from-scratch and pre-trained models.
* They foster a collaborative and open-source ethos. It provides a platform where researchers and engineers worldwide can share, improve, and reuse models and code. This collaborative spirit eliminates redundancy and accelerates progress in the field of AI. By fostering this ecosystem, we contribute to more sustainable AI research practices.

[next]

## Software and Environment:

* The heart of this project is Python and the Google Colab environment. This provides access to essential computing resources like GPUs and additional disk space. While these resources are free, it's worth noting that they have usage limits, which refresh after an unspecified duration.
* In the Colab environment, the decision between PyTorch and TensorFlow ML libraries was influenced by practical considerations. Colab comes pre-installed with some TensorFlow packages, simplifying the development process.
* Moreover, TensorFlow’s support for Accelerated Linear Algebra (XLA) significantly accelerates the compilation of our transformer models. This performance boost enhances both the development speed and overall efficiency of our project.

[next]

## Harnessing the Tokenizers:

## Hugging Face allows us to load the PEGASUS and BART tokenizers using the same checkpoint names. The are used to process our dataset in conjunction with a data collator that implements batch processing with a batch size of 2.

[next]

## Fine-Tuning for Precision:

## When it's time to fine-tune our pre-trained models for their specialized task: We deploy an AdaFactor optimizer employing a custom learning rate scheduler that changes it from 6x10-4 to 6x10-5 after 7 epochs. The inclusion of XLA in our arsenal significantly reduces compilation time, optimizing our training process – wherein each model undergoes a 10-epoch training cycle. After each epoch, Hugging Face allows us to use a unique type of callback that can save the model in my Hugging Face repository.

[next]

## Testing the Waters:

## To test our models, we turn to the first 100 lyric-interpretation pairs from the test set of the Song Interpretation Dataset. The team behind BART-fusion has crafted it with manually, such that it avoids any overlaps with the training set, ensuring that our models face a true challenge. This testing environment eliminates any chances of data leakage (a situation where the model has already come across the test data). To generate the predictions, we again employ an XLA function, this time to improve the speed at which the model generates these tokens.

[next]

## Evaluation metrics:

We use the same metrics as the team behind the BART-fusion model, namely:

Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

This set of metrics measure the amount of critical information retained in the generated summary when compared to a human based one. Each metric is divided into recall, precision, and f1-measures. These represent how well a generated sentence matches the target sentence, but each paints a slightly different picture.

Recall is number of overlapping words divided by total words in target sentence. Precision refers to the ratio of the number of overlapping words to the total words in generated summary (essentially indicates how much of the generated summary was relevant). F1-measure incorporates both recall and precision to give us a more complete view of how good the summary is.

ROUGE has several different scores. We consider ROUGE-1, ROUGE-2 (measures unigram and bigram overlap respectively) and ROUGE-L (measures the longest subsequence of words).

METEOR score

Mainly used for evaluating machine translation based on the idea of unigram/singular-word matching . Note that these matches are calculated in various forms such as surface, stemmed and most importantly meaning (in other words, synonyms can also be matched). A singular score is then calculated from the recall, precision, and f-measure of these matches.

In the context of translation, it designed to indicate how well the word order of a translation matches the target sentence. However, we can leverage its ability to capture matches for similar words – in contrast to ROUGE, which only considers the given words.

BERTScore

The BERTScore matches words in the generated and human sentences via a cosine similarity (note that it also utilizes recall, precision, and f-measure values). The idea is to provide an insight into as to the semantic similarity of the sentences. It has been demonstrated to be close to human-level judgement in terms of evaluation.

Having a good interpretation is not worth much if it is not a natural and fluent example of the English lexicon. Hence, we consider BERTScore as one of the main metrics – considering it equally important to those metrics evaluating the quality of the summary.

Note that the research teams behind BART had access to 64 TPUs and their hardware could train their models over several days. When compared to using an environment like Google Colab - although a powerful – is ultimately meant for use by students and casual members of the AI community. Thus, the results are predictably inferior to those obtained by research teams. However, it can be demonstrated that this project can achieve similar results using extrapolation. That is, I train the models on increasingly larger subsets of the dataset and show that the results and metrics are directly proportional.

# Results

The following graphs show the resulting metrics achieved by the PEGASUS and BART models (shown by blue and red respectively). Note that only the f-measure scores are shown (as they are a more accurate representation compared to recall and precision measures).

Although the BART model has achieved higher scores, we can observe that the PEGASUS model has a steeper gradient across all metrics. This implies that, it learns more from the increases in data compared to the BART model.

We can also note that PEGASUS took longer to train than BART, except in the case of 3000 songs.

Ultimately, the models do not hold a candle to the results from BART-fusion, which outperforms us by a large margin. This includes even the BART model used as a baseline to test their BART-fusion model. This is a direct consequence of attempting a task of this magnitude at honors level (with the resources to match).

[next]

However, we need not despair. As hypothesized, we can notice a steady increase in the metrics as more data is fed into the models. The data displays evidence of bring a polynomial of degree one, i.e.: it has a (near) linear relationship between the number of songs loaded and the metrics. If we perform extrapolation to find the values when the full dataset is loaded, we get:

Trying to extrapolate the metrics when 300,000 songs are loaded from data has only 3000 is widely out of scope (observe that some metrics are decreasing as more songs are added!) and it should be noted that it is the nature of extrapolation to be less accurate the further we want to predict. Hence, these graphs and values are not accurate and do not represent any conclusive results that would have been obtained if the models were able to train fully.

[next]

# While our results may not have reached the pinnacle of perfection, we must acknowledge the constraints we faced, particularly in terms of resources. The field of song lyric analysis is a challenging one, and as a lone explorer with limited resources, we faced formidable obstacles. However, this serves as a testament to the potential that lies ahead with even greater support and resources. Our journey is far from over, and our current findings pave the way for future improvements and accomplishments.

# Future work

In future work, while it is an important addition to the task of lyric interpretation, the Song Interpretation Dataset itself can be improved. Specifically, when it comes to the interpretations themselves. The recommendations I have in mind include:

1. **Enhanced Dataset Quality:** Improve the dataset quality by curating interpretations that provide in-depth analysis similar to the scholarly study of poetry. This would offer models a more comprehensive understanding of song lyrics by associating words with diverse concepts beyond the lyrics' explicit content.
2. **Feature Extraction:** Implement feature extraction from song lyrics, including rhyme structures, orientation, and other linguistic elements, to enhance model understanding. Building on previous work that classifies music into genres based on lyric features, these features could provide valuable insights.
3. **Incorporate Artist and Context Data:** Expand the dataset to include information about the song's artist(s) and the historical context in which it was written. This additional data can contribute to a deeper interpretation of lyrics and their intended meaning.
4. **Experiment with Diverse Pre-trained Models:** Given more substantial computing resources, experiment with a wider array of pre-trained models beyond BART and PEGASUS to discover which architecture performs best for song lyric interpretation. This could include investigating the latest models developed in the field.