Over the past few days I have been learning how to web scrape using BeautifulSoup package, create word and sentence embeddings using 🤗, and compare these embedding using cosine similarity. Below I am going to write psuedo code that will realize and MVP for my model, so I can see what is lacking, and what is really cool.

*Web Scraping*

Use BeautifulSoup to scrape the 112 query pages of [jkrishnamurti](https://jkrishnamurti.org/jksearch?keyword=&page=112&type=16618) archive for links to quotes

Use BeautifulSoup scrape each webpage for the quote. Append each quote to the end of a “quotes” list

Use Pandas to create dataframe of all 1100 quotes

-------------------------------------------------------------------------------------------------------------------------------

*Sentence2Vec*

Find a model that can take in a short query, (What is the meaning of life?)

And find relevant answers (quotes)

We will use a asymmetric model as answers > queries

Write a function that creates an embedding of each quote

1. Split up quote into sentences
2. Feed sentences into sentence transformer
3. Append embedding to list of quote embeddings
4. Add ‘embeddings’ column to dataframe to store representation

-------------------------------------------------------------------------------------------------------------------------------

*Gradio*

Create a user interface in which a user enters a question,

And both text generated and a dataframe of quotes are returned to them

1. Create framework of the gradio UI (set up the Interface class instance)
2. Write a functions that returns
   1. Text generated from query
   2. Similar quotes to the query

-------------------------------------------------------------------------------------------------------------------------------

Pushing model to 🤗 Spaces

Here I want to make this model accessible to anyone on the Internet by designing a Hugging Face Space

1. Add requirement.txt file to specify python dependencies
2. Create app.py file with gradio code