From Text to Visuals: Leveraging AI for Immersive Storytelling(Edgar Allen Poe) Final Paper

1. Abstract

This research explores the potential of artificial intelligence (AI) to create seamless and customizable visual representations of literary texts, specifically focusing on Edgar Allan Poe's works. The aim is to bridge the gap between traditional literature and modern digital media by enhancing readers' visualization of narratives through AI-generated imagery. This is particularly important in the digital age, where engagement with text is evolving due to shorter attention spans and the rise of multimedia content. The methodology involved data preprocessing, including text normalization, tokenization, and sentiment analysis, followed by genre classification using natural language processing(NLP) techniques and a Random Forest classifier. Advanced NLP and image generation models, GPT-3.5-Trubo and DALL-E were then used to create detailed and consistent visual representations of the text. The result demonstrated the models' ability to generate images that aligned with the narrative and tone of the text, though challenges with consistency and handling complex narrative elements were identified. The study concludes that with further refinement, AI has the potential to transform the reading experience by offering new opportunities for engagement , accessibility, and innovation in the literary space. Future research will focus on experimenting with more advanced models, expanding the dataset, and exploring techniques for maintaining visual consistency across generated images.

2. Introduction

This question is important because it explores the background and intersection of technology and literature, potentially paving the way for new forms of interactive and immersive reading experiences. As a person with an avid reading addiction for both classical literature as well as webcomics, I felt this to be a perfect way to combine my passions. As digital media evolves, readers also increasingly seek ways to enhance their engagement with texts, as the average attention span has dropped from 12 seconds to 8 seconds from 2000 to 2015, not putting into play the last 9 years(, that have had a increasingly rapid increase in short form content, and AI-driven visual representations could offer a novel approach to this

Benefits to the Reading Space:

Enhanced Engagement: By providing readers with customizable visuals that align with their imagination, the reading experience can become more immersive, drawing readers deeper into the narrative and potentially increasing comprehension and enjoyment.

Accessibility: For readers who struggle with visualizing text, such as those with cognitive disabilities or reading difficulties, AI-generated imagery could serve as a supportive tool, making literature more accessible and enjoyable.

Personalization: The ability to customize visuals allows readers to tailor their experience to their preferences, creating a more personalized and meaningful interaction with the text.

Innovation in Publishing: This approach could lead to new formats for e-books and other digital publications, offering a fresh way to consume literature that goes beyond traditional text and static images.

Expanding the Market: By offering a unique reading experience, publishers and authors might attract new audiences who are more accustomed to visual media, bridging the gap between traditional literature and the digital age.

Overall, this research has the potential to transform the way people interact with literature, offering new opportunities for engagement, accessibility, and innovation in the reading space.

To do this I had to identify that this problem was supervised and that it was a classification model, and I was working on a natural language model, to convert pages in books into effective sentences to put into an Ai image generator.

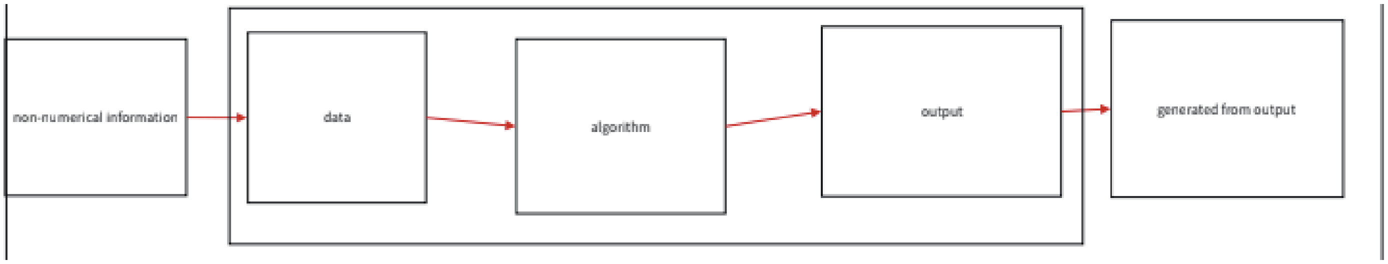


Illustration of the Machine learning pipeline

3. Background

There haven't been many articles or approaches to solve my research. On the other hand, there are a lot of articles that correlate with AI and imagery and LLMs that can help solve the research problem, some examples include:

1) ***Sentiment Analysis Techniques and their application in Creative AI***

One approach that has been influential in the field of AI is the application of sentiment analysis (SA) techniques. These methods have been used to understand and classify emotions and sentiments in text, which can be indirectly related to generating corresponding imagery for literary content. Sentiment analysis involves categorizing text into different sentiment categories and extracting relevant features that may be used to inform image generation. For example, if a particular page of a book evokes strong emotions of sadness, an AI could use these cues to generate an image with corresponding themes and colors. The pros of this is that sentiment analysis can provide the contextual emotional cues that can enhance the accuracy and relevance of the generated images, making them more reflective of the text's emotional tone. On the other hand the cons, is that Sentiment analysis is still in the early stage of development, when it comes to handling domain-specific and contradiction-based analysis, which could result in less nuance or even misleading visual representations. This dataset has the unique contribution of providing emotional context, but will need to be combined with other techniques to fully translate complex narratives into images.

2) ***AI in Art Generation: Democratizing creativity***

Another approach that directly relates to my project is the use of AI for art generation through tools such as DALL-E, Midjourney, and Stable Diffusion. These platforms have revolutionized the way art is created, allowing for images to be generated based on textual prompts. This technology is important for my research because it shows the technical feasibility to convert text into images. The pros of this is that AI art generators have been proven successful in translating prompts into detailed and imaginative visuals. The accessibility of these tools makes them an excellent starting point for developing a more specialized system for book-to-image translation. On the other hand, there are cons, while the art can be effective, the art has a hard time being consistent across a narrative, and ensuring that images align with the specific details and style of a book, which is very important for this experiment. The unique contribution is that there is already a large space of text-to-image translation, setting a solid foundation. However, it stresses the importance of focusing on narrative consistency and capturing the detailed subtleties of each page.

3) Large Language Models (LLMs) as Advanced Text processors

Large Language Models (LLMS) like Chat Gpt and BERT have shown extraordinary abilities of processing and synthesizing text. The understanding of visuals and generating corresponding visuals is still being developed, but it could be integrated into my machine learning pipeline to assist in understanding the narrative structure of a book, thereby guiding the image generation process. On the other hand the negative, of the LLMS true comprehension of text and may miss the more nuanced meaning of the text, is concern, as well as not being able to replace the true capacity of human creativity in traditional art. The unique contribution of a LLMs is that it would be able to create an enhancement of accuracy of the texts analysis, helping to illustrate the appropriate context that aligns with the narratives flow.

These approaches provide a diverse set of tools and techniques that can be combined and adapted to your research problem of translating books into immersive visual experiences. By leveraging sentiment analysis for emotional context, AI art generation tools for creating visual content, and LLMs for understanding narrative structure, your project can address the challenges of consistency, nuance, and creativity in book-to-image translation, in an ideal setting.

4. Dataset

For this project, the dataset used comprises the full corpus of 69 short stories written by Edgar Allan Poe. This dataset is essential for the Natural Language Processing (NLP) tasks required in translating the text into images, page by page, for an immersive reading experience.

Type of Data

The dataset includes **textual data**, with each short story represented as a text document. Alongside the text, there are **classification labels** that categorize the stories based on genre and other features, such as publication date. Additionally, the dataset includes numerical vectors derived from text processing techniques, which are used as input features for machine learning models.

Number of Samples

The dataset consists of 69 samples(books), each corresponding to one of Poe's short stories. Each sample includes the following features:

- **Text**: The full text of the short story.
- **Classification**: A label that classifies the story by genre.
- **Publication Date**: The date when the story was originally published.
- **Processed Text**: A cleaned and tokenized version of the text, ready for machine learning processing.
- **Average Word Vector**: A numerical representation of the text derived from the Skip-gram model.

Data Preprocessing

The preprocessing of the dataset involved several steps:

1. **Text Cleaning**: The text of each story was converted to lowercase, and punctuation was removed to ensure uniformity.
2. **Tokenization**: The cleaned text was then tokenized into individual words.
3. **Stopword Removal**: Common English stop words were removed to reduce noise and focus on meaningful content.
4. **Text Vectorization**: The processed text was transformed using the Skip-gram model from Word2Vec, generating vector representations for each story based on the contextual relationships between words.
5. **Label Encoding**: The classification labels were encoded numerically for compatibility with machine learning algorithms.
6. **Handling Rare Classes**: Classes with fewer than two samples were excluded to ensure the model had enough data to learn meaningful patterns.

Data Splitting

The dataset was split into **training and testing sets** using an 80-20 split, with stratification based on the classification labels to ensure a balanced distribution of classes in both sets. The training set was used to train the machine learning model, while the testing set was used to evaluate the model's performance.

Data Visualization

To better understand the dataset and its features, several visualizations were created:

1. **Word Frequency Analysis**: For each classification category, the top 10 most frequent words were identified and plotted. This visualization helps in understanding the common themes and language used in different genres of Poe's stories.
2. **Word Vector Distribution**: The average word vectors for the stories were visualized to explore the spatial distribution of the stories in the vector space. This can reveal clusters of stories that are similar in theme or language.
3. **Classification Distribution**: A bar chart showing the distribution of stories across different classification labels was generated to ensure that the dataset was balanced.

Significance of Features

- **Processed Text**: This is the core feature that drives the translation of text into images. By cleaning and tokenizing the text, the dataset becomes suitable for NLP and machine learning models.
- **Average Word Vector**: These vectors capture the semantic meaning of the stories and are crucial for ensuring that the generated images are contextually relevant.
- **Classification Labels**: These labels help in categorizing the stories, which can be useful for genre-specific image generation and ensuring consistency in visual style

Overall, the dataset provides a rich source of textual information and features that are essential for developing a machine learning pipeline capable of translating literature into immersive visual experiences

5. Methodology/Models

Hugging Face Generation Attempt

This attempt was in hope of creating a system that can generate vivid and accurate imagery descriptions from book text.

The model selection was GPT-2, a generative pre=trained transformer model, which is particularly well suited for text generation. Chat-GPT's architecture is based on transformers, a deep learning model that excels in handling sequential data, making it ideal for understanding and generating human-like text.

Data Preparation and preprocessing

The first step involved gathering a dataset of short stories with image descriptions. This dataset(E.A. Poe’s corpus of short stories) served as a training ground for the model to learn how to associate textual descriptions with visual imagery.

E.A. Poe’s corpus of short stories, a CSV file contained story texts. The dataset was curated with the thought in mind of a multifaceted variety of genres and tones to train the model in distinguishing genre and tone.

The data was prepared combining the title, genre, and story text into a single input string. This was done to give the model context about the story's setting and tone, which are critical to generating accurate imagery.

```
# Combine title, text, and genre into a single input string for the model

story_data['input'] = story_data.apply(

    lambda row: f"Title: {row['title']} Genre: {row['classification']} Story:
{row['text']}", axis=1

)

story_data.head()
```

The model also used data splitting -into training and validation sets using an 80/20 ratio. The training set was used to fine tune the model, while validation was used to set its performance.

```
# Split data into training and validation sets
train_df, val_df = train_test_split(story_data, test_size=0.2)

train_texts = train_df['input']
train_labels = train_df['image description']
val_texts = val_df['input']
val_labels = val_df['image description']
```

The next step was tokenization and embedding, before training the data. Tokenization is the process of converting text into numerical data that the model can understand. Special tokens, such as padding tokens, were added to ensure all sequences had the same length.

```
tokenizer.add_special_tokens({'pad_token': '[PAD]'})
model.resize_token_embeddings(len(tokenizer))  # Update the model's
token embeddings
```

Acustum PyTorch dataset class was also created to manage and train the validation data effectively during the training process

```
class StoryDataset(torch.utils.data.Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels
```

To fine tune the model we used the Hugging Face ‘Trainer’ class. The fine tuning process involved adjusting the model’s weights to improve its ability to generate accurate and contextually rich image descriptions based on the input text.

The training setup was configured with special arguments, which included: number of epochs, batch size, learning rate, and logging steps. Which was designed to optimize the learning process without overfitting.

```
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=2,
    per_device_train_batch_size=2,
    per_device_eval_batch_size=2,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    logging_steps=10,
)
```

The training was monitored on the validation site. The ‘Trainer’ class provided an efficient way to manage the training loop, handle evaluation, and save the model checkpoints.

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
)

trainer.train()
```

After training was done, the model would have generated image descriptions and new story inputs. This step involved feeding the model a text prompt and generating an output that described the scene in detail for the Chat-gpt 2 model.

```
def generate_description(story_input):
    inputs = tokenizer.encode(story_input, return_tensors='pt')
    outputs = model.generate(inputs, max_length=50,
num_return_sequences=1, pad_token_id=tokenizer.eos_token_id)
    return tokenizer.decode(outputs[0], skip_special_tokens=True)

# Example usage
```

```
example_story_input = "Title: Example Title Genre: Fantasy Story: This
is an example of a fantasy story text."
description = generate_description(example_story_input)
print(description)
```

Problem/conclusion

The problem with this model was the amount of training time, due to the tokenization length being the same for every single token, leading to an excessive time to process that wouldn't work on my computer. But in theory this process could be effective. Although the Chat gpt 2 model is outdated, other models could lead to a more theoretical optimized and better result.

Chat Gpt 3.5 Turbo model

This was the main model in the research, utilizing this attempt to get to understand NLP's and data preprocessing while also getting a viable product.

For this model, I first used E.A Poe's Corpus as a dataset for my data preprocessing, genre classification, and NLP's. This model helped me learn how to understand data.

Data Preprocessing is crucial for preparing textual data for machine learning analysis, which is why it's important to understand. I utilized loading and inspecting data, text normalization, tokenization and stopword removal, sentiment analysis, and word frequency analysis. These preprocessing steps helped to facilitate feature extraction, and improve data interpretability, making them essential for understanding complex literary corpora like Edgar Allen Poe's works. I also utilized genre classification, through NLP models(SpaCy 'en_core_web_md' model is loaded for advanced natural language processing tasks and necessary NLTK resources like stop words are downloaded. I also used custom tokenization, to tokenize the process, by removing stopwords, and filtering out pronouns, and lemmatizing words(reducing words to base form). This step is essential for preparing for genre classification. I then used 'CountVectorizer' to convert the processed text into a bag of words model. Thereby, reducing the text to numerical features, which can be used for machine learning models. This feature is limited to 800 to avoid overfitting and generalizability. Lastly, I used data splitting similar to my HuggingFace attempt with a 80-20 split. Allowing the model to learn from a portion of the data and be evaluated on unseen data, ensuring a robust genre classification model. One problem with this model was getting only one genre classification per story, but wasn't that impactful due to my data, because I just wanted to understand how genre classification works, and the importance of generating genre specific content in images. Lastly I made a random forest classifier, the significance of this model was that it allows for a robust and interpretable genre classification by capturing Word2VEc embeddings. It is particularly effective for tasks involving textual data, like Edgar Allens Poe's corpus, which relies on a semantic nauseous method that is crucial for accurate classification.

With the 69 short stories this model was well-suited for analyzing the complex literary dataset.

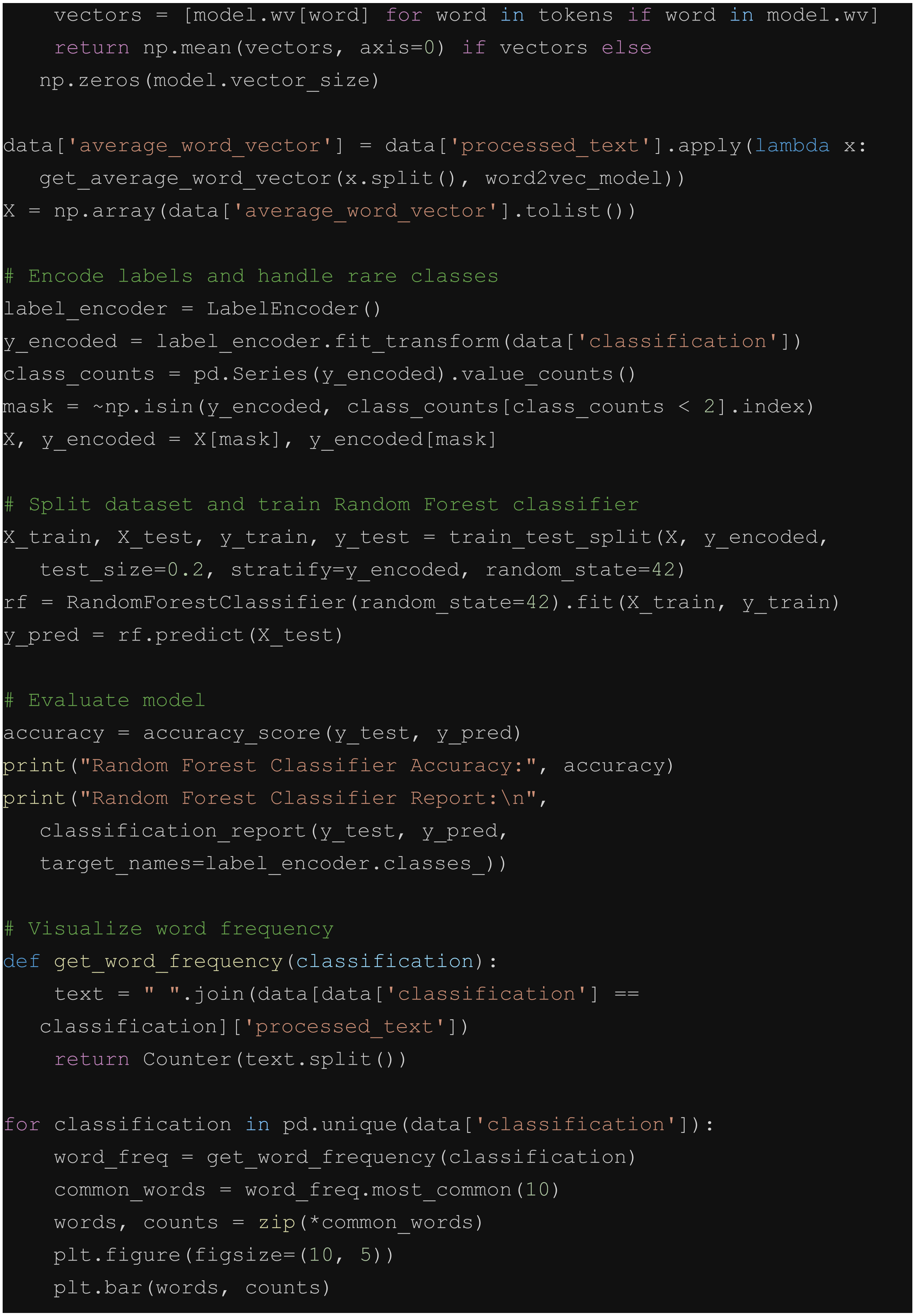
```
import pandas as pd
import numpy as np
import re
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from string import punctuation
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from collections import Counter
import matplotlib.pyplot as plt
from gensim.models import Word2Vec

# Download NLTK data
nltk.download('stopwords')
nltk.download('punkt')

# Load and preprocess data
data =
   pd.read_csv("https://raw.githubusercontent.com/IanRmacdonnell/Short-
   story-storage/main/preprocessed_data.csv")
data['processed_text'] = data['text'].apply(lambda x: " ".join(
    [word for word in
   word_tokenize(re.sub(f"[{re.escape(punctuation)}]", "", x.lower()))
   if word not in set(stopwords.words('english'))]
))

# Train Skip-gram model
sentences = data['processed_text'].apply(lambda x: x.split()).tolist()
word2vec_model = Word2Vec(sentences, vector_size=100, window=5, sg=1,
   min_count=1, workers=4)
word2vec_model.save("skipgram.model")

# Transform text data using Skip-gram model
def get_average_word_vector(tokens, model):
```



```
    vectors = [model.wv[word] for word in tokens if word in model.wv]
    return np.mean(vectors, axis=0) if vectors else
   np.zeros(model.vector_size)

data['average_word_vector'] = data['processed_text'].apply(lambda x:
   get_average_word_vector(x.split(), word2vec_model))
X = np.array(data['average_word_vector'].tolist())

# Encode labels and handle rare classes
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(data['classification'])
class_counts = pd.Series(y_encoded).value_counts()
mask = ~np.isin(y_encoded, class_counts[class_counts < 2].index)
X, y_encoded = X[mask], y_encoded[mask]

# Split dataset and train Random Forest classifier
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded,
   test_size=0.2, stratify=y_encoded, random_state=42)
rf = RandomForestClassifier(random_state=42).fit(X_train, y_train)
y_pred = rf.predict(X_test)

# Evaluate model
accuracy = accuracy_score(y_test, y_pred)
print("Random Forest Classifier Accuracy:", accuracy)
print("Random Forest Classifier Report:\n",
   classification_report(y_test, y_pred,
   target_names=label_encoder.classes_))

# Visualize word frequency
def get_word_frequency(classification):
    text = " ".join(data[data['classification'] ==
   classification]['processed_text'])
    return Counter(text.split())

for classification in pd.unique(data['classification']):
    word_freq = get_word_frequency(classification)
    common_words = word_freq.most_common(10)
    words, counts = zip(*common_words)
    plt.figure(figsize=(10, 5))
    plt.bar(words, counts)
    plt.title(f"Top 10 Words in {classification}")
    plt.show()
```

All of this was used to understand how LLM works and how a machine pipeline works. But the problems with this process similar to the Hugging Face model is the complexity and nuance.

So instead I used this knowledge to utilize an established and advanced NLP and image generation models-Chat gpt 3.5 turbo and Dall-E through an API key to make a more nuanced imagery of stories plausible. This problem still came with the problem of setting up NLP tools, generating requirements to make the text prompts detailed enough to create a strong theme among images from page to page, while also handling API rate limits, due to the 5 request per minute limit.

```
!wget -q --show-progress "https://raw.githubusercontent.com/IanRmacdonnell/Short-story-storage/main/preprocessed_data.csv"
!wget -q --show-progress "https://raw.githubusercontent.com/IanRmacdonnell/Short-story-storage/main/Short_Stories_with_Image_Descriptions.csv"
#!pip install openai==0.28
!pip install openai
!pip install spacy
!python -m spacy download en_core_web_sm

#create a chatgpt key- use files- api key- open ai google collab
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
import spacy
from openai import OpenAI
from IPython.display import Image, display
import time

# Initialize spaCy and OpenAI API
nlp = spacy.load("en_core_web_sm")
client = OpenAI(api_key="PUT YOUR OWN API KEY")

# Author and title of the book
```

```python
author = "Edgar Allen Poe"
title = "The man of the Crowd"

# Prompt to generate the summary, character list, tone, genre, and page-by-page imagery descriptions
prompt = f"""Can you summarize the plot of the book titled '{title}' by {author}?
Please include the following:
1. A summary of the plot.
2. A list of all the main characters with descriptions.
3. The tone and genre of the book.
4. Imagery descriptions for each page of the book, ensuring they include relevant details such as the tone, genre, and character appearances for the most accurate imagery.
5. Ensure the imagery descriptions are suitable for generating images with no text or words included in the visuals.
6. Do page numbers
7.Create a setting before each image, and make sure the imagery is not completely different between images"""

# Model choice
model_choice = "gpt-3.5-turbo"

# Function to generate text using OpenAI API
def generate_text(prompt, model=model_choice):
    try:
        response = client.chat.completions.create(
            model=model,
            messages=[
                {"role": "system", "content": "You are a hardworking assistant."},
                {"role": "user", "content": prompt}
            ]
        )
        return response.choices[0].message.content.strip()
    except Exception as e:
        return f"An error occurred: {e}"

# Generate the text
generated_text = generate_text(prompt)
```

```
print("Generated Text:")
print(generated_text)

# Extract summary, characters, and imagery descriptions
summary_start = generated_text.find("Summary:") + len("Summary:")
characters_start = generated_text.find("Characters:")
imagery_start = generated_text.find("Imagery Descriptions:")

summary = generated_text[summary_start:characters_start].strip()
characters = generated_text[characters_start:imagery_start].strip()
imagery_descriptions = generated_text[imagery_start + len("Imagery Descriptions:"):].strip()

# Process the imagery descriptions into separate pages
doc = nlp(imagery_descriptions)
pages = [sent.text.strip() for sent in doc.sents]

print("Summary:")
print(summary)
print("\nCharacters:")
print(characters)
print("\nPage Descriptions (will only display the current page during generation):")

# Function to generate an image with error handling and rate limit management
def generate_image_with_retry(prompt, retry_limit=3):
    for attempt in range(retry_limit):
        try:
            response = client.images.generate(
                model="dall-e-3",
                prompt=prompt,
                size="1024x1024",
                quality="standard",
                n=1,
            )
            return response.data[0].url
        except Exception as e:
            error_message = str(e)
            if "rate_limit_exceeded" in error_message:
                print("Rate limit exceeded. Waiting for 10 seconds before retrying...")
                time.sleep(10)
            else:
                print(f"An error occurred: {e}")
                break
    return None

# Loop through each page description and generate an image
for i, page_description in enumerate(pages, 1):
    image_prompt = f"{summary}\n{characters}\nPage {i} description: {page_description}\nNote: No text or words should be present in the image."
    print(f"Generating image for Page {i} with prompt:\n{page_description}")

    # Generate the image with retry mechanism
    image_url = generate_image_with_retry(image_prompt)

    if image_url:
        # Print the URL for debugging purposes
        print(image_url)
        # Display the generated image
        display(Image(url=image_url))
        # Display the text of the page
        print(f"Page {i} Text: {page_description}")
    else:
        print(f"Failed to generate image for Page {i} after retries.")

    # Wait to ensure we don't exceed the rate limit
    time.sleep(12)  # 12 seconds between requests to stay under the 5 requests/minute limit
```

6. Results

This is the created images of the first 9 pages of The Man of the Crowd through the prompts: The man of the Crowd, fittingly by Edgar Allen Poe.

Generated Text:

Summary of the Plot:
"The Man of the Crowd" by Edgar Allan Poe is a short story that follows an unnamed narrator who observes a man from the window of a London coffee house as he wanders aimlessly through the crowded streets. The narrator becomes intrigued by the man's mysterious and erratic behavior, leading him to follow the man through the bustling city for hours. Despite the narrator's efforts, he is unable to discern the man's true intentions or identity, leaving him unsettled and perplexed by the enigmatic figure he has been watching.

Main Characters:
1. The Narrator: An unnamed individual who becomes fixated on observing the man of the crowd.
2. The Man of the Crowd: A mysterious and elusive figure whose actions captivate the narrator.

Tone and Genre:
The tone of "The Man of the Crowd" is dark, mysterious, and suspenseful. The genre can be classified as a psychological thriller or mystery.

Imagery Descriptions:
Page 1 - Setting: Victorian-era London street crowded with people. The narrator watches from a coffee house window, his gaze fixed on the solitary figure of the man of the crowd, who seems to blend into the throngs of passersby with his nondescript appearance.

Page 2 - Setting: Narrow alleyway with dimly lit gas lamps casting long shadows. The man of the crowd moves swiftly through the labyrinthine streets, his features obscured by the darkness, adding to his air of mystery and intrigue.

Page 3 - Setting: Rain-soaked cobblestone street reflecting the eerie glow of flickering streetlamps. The man of the crowd hurries through the wet thoroughfare, his silhouette distorted by the shimmering puddles, heightening the sense of tension and uncertainty surrounding his enigmatic presence.

Page 4 - Setting: Damp and fog-shrouded city square enveloped in an eerie silence. The man of the crowd pauses momentarily, his back turned to the narrator, creating an unsettling image of a lone figure in a desolate urban landscape, reinforcing the story's ominous atmosphere.

Let me know if you need more pages or further details.

Summary:
of the Plot:
"The Man of the Crowd" by Edgar Allan Poe is a short story that follows an unnamed narrator who observes a man from the window of a London coffee house as he wanders aimlessly through the crowded streets. The narrator becomes intrigued by the man's mysterious and erratic behavior, leading him to follow the man through the bustling city for hours. Despite the narrator's efforts, he is unable to discern the man's true intentions or identity, leaving him unsettled and perplexed by the enigmatic figure he has been watching.

Main

Characters:

Characters:
1. The Narrator: An unnamed individual who becomes fixated on observing the man of the crowd.
2. The Man of the Crowd: A mysterious and elusive figure whose actions captivate the narrator.

Tone and Genre:
The tone of "The Man of the Crowd" is dark, mysterious, and suspenseful. The genre can be classified as a psychological thriller or mystery.

Page Descriptions (will only display the current page during generation):
Generating image for Page 1 with prompt:
Page 1 - Setting: Victorian-era London street crowded with people.
https://oaidalleapiprodscus.blob.core.windows.net/private/org-JIQuPumgUyJ9ld7QC8n1tutN/user-o2bO6Ni1LRgEDWJ3J1ALHbB5/img-ONfl70zvgAY8Mw1JSUDJrTqn.png?st=2024-08-31T18%3A13%3A26Z&se=2024-08-31T20%3A13%3A26Z&sp=r&sv=2024-08-04&sr=b&rscd=inline&rsct=image/png&skoid=d505667d-d6c1-4a0a-bac7-5c84a87759f8&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-08-30T23%3A25%3A58Z&ske=2024-08-31T23%3A25%3A58Z&sks=b&skv=2024-08-04&sig=xhiLh8fZBWRT8dxU9JxtUiUFanPuz6okO/2/%2BrVJOf0%3D

Page 1 Text: Page 1 - Setting: Victorian-era London street crowded with people.

Generating image for Page 2 with prompt:

The narrator watches from a coffee house window, his gaze fixed on the solitary figure of the man of the crowd, who seems to blend into the throngs of passersby with his nondescript appearance.

https://oaidalleapiprodscus.blob.core.windows.net/private/org-JIQuPumgUyJ9ld7QC8n1tutN/user-o2bO6Ni1LRgEDWJ3J1ALHbB5/img-k7mRRCifodQtDZZAUop458jP.png?st=2024-08-31T18%3A13%3A49Z&se=2024-08-31T20%3A13%3A49Z&sp=r&sv=2024-08-04&sr=b&rscd=inline&rsct=image/png&skoid=d505667d-d6c1-4a0a-bac7-5c84a87759f8&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-08-31T00%3A22%3A44Z&ske=2024-09-01T00%3A22%3A44Z&sks=b&skv=2024-08-04&sig=I4cT34a1H58tx8UuanT4UkgiHIcppg8iByUamoaklq0%3D

Page 2 Text: The narrator watches from a coffee house window, his gaze fixed on the solitary figure of the man of the crowd, who seems to blend into the throngs of passersby with his nondescript appearance.

Generating image for Page 3 with prompt:

Page 2 - Setting: Narrow alleyway with dimly lit gas lamps casting long shadows.

https://oaidalleapiprodscus.blob.core.windows.net/private/org-JIQuPumgUyJ9ld7QC8n1tutN/user-o2bO6Ni1LRgEDWJ3J1ALHbB5/img-lA27x8yRECAh0KZcLp0mQ8WP.png?st=2024-08-31T18%3A14%3A13Z&se=2024-08-31T20%3A14%3A13Z&sp=r&sv=2024-08-04&sr=b&rscd=inline&rsct=image/png&skoid=d505667d-d6c1-4a0a-bac7-5c84a87759f8&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-08-30T23%3A18%3A55Z&ske=2024-08-31T23%3A18%3A55Z&sks=b&skv=2024-08-04&sig=thcDTxqZAidL49Ayuxw9fHCxOz1hf17nzuiIQIZlAyc%3D

Page 3 Text: Page 2 - Setting: Narrow alleyway with dimly lit gas lamps casting long shadows.
Generating image for Page 4 with prompt:
The man of the crowd moves swiftly through the labyrinthine streets, his features obscured by the darkness, adding to his air of mystery and intrigue.
https://oaidalleapiprodscus.blob.core.windows.net/private/org-JIQuPumgUyJ9ld7QC8n1tutN/user-o2bO6Ni1LRgEDWJ3J1ALHbB5/img-1dVBA8KBDvWqvgPsDgEF9HwF.png?st=2024-08-31T18%3A14%3A41Z&se=2024-08-31T20%3A14%3A41Z&sp=r&sv=2024-08-04&sr=b&rscd=inline&rsct=image/png&skoid=d505667d-d6c1-4a0a-bac7-5c84a87759f8&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-08-30T23%3A12%3A42Z&ske=2024-08-31T23%3A12%3A42Z&sks=b&skv=2024-08-04&sig=2RFEympV1EKIkL2NmsPTasv8PuUKnXb4HO1P9/uHa9c%3D

COF FFEE CHOUE

Page 4 Text: The man of the crowd moves swiftly through the labyrinthine streets, his features obscured by the darkness, adding to his air of mystery and intrigue.

Generating image for Page 5 with prompt:

Page 3 - Setting: Rain-soaked cobblestone street reflecting the eerie glow of flickering streetlamps.

https://oaidalleapiprodscus.blob.core.windows.net/private/org-JIQuPumgUyJ9ld7QC8n1tutN/user-o2bO6Ni1LRgEDWJ3J1ALHbB5/img-BW0meCwZjqYEgJ1wYUDOphbk.png?st=2024-08-31T18%3A15%3A05Z&se=2024-08-31T20%3A15%3A05Z&sp=r&sv=2024-08-04&sr=b&rscd=inline&rsct=image/png&skoid=d505667d-d6c1-4a0a-bac7-5c84a87759f8&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-08-30T23%3A37%3A01Z&ske=2024-08-31T23%3A37%3A01Z&sks=b&skv=2024-08-04&sig=t1AYQWbnccvHZeAE0MHT5tqj4tEtp7qK7nN1K5%2BRpQ8%3D

Page 5 Text: Page 3 - Setting: Rain-soaked cobblestone street reflecting the eerie glow of flickering streetlamps.

Generating image for Page 6 with prompt:

The man of the crowd hurries through the wet thoroughfare, his silhouette distorted by the shimmering puddles, heightening the sense of tension and uncertainty surrounding his enigmatic presence.

https://oaidalleapiprodscus.blob.core.windows.net/private/org-JIQuPumgUyJ9ld7QC8n1tutN/user-o2bO6Ni1LRgEDWJ3J1ALHbB5/img-XGCqSrlntwRPde43O1lOevJ8.png?st=2024-08-31T18%3A15%3A31Z&se=2024-08-31T20%3A15%3A31Z&sp=r&sv=2024-08-04&sr=b&rscd=inline&rsct=image/png&skoid=d505667d-d6c1-4a0a-bac7-5c84a87759f8&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-08-30T23%3A25%3A50Z&ske=2024-08-31T23%3A25%3A50Z&sks=b&skv=2024-08-04&sig=099xX7JGYxK%2BKRw08Gw1MbvJIupXvfwMgH5OvUR9Ytc%3D

Page 6 Text: The man of the crowd hurries through the wet thoroughfare, his silhouette distorted by the shimmering puddles, heightening the sense of tension and uncertainty surrounding his enigmatic presence.

Generating image for Page 7 with prompt:

Page 4 - Setting: Damp and fog-shrouded city square enveloped in an eerie silence.

https://oaidalleapiprodscus.blob.core.windows.net/private/org-JIQuPumgUyJ9ld7QC8n1tutN/user-o2bO6Ni1LRgEDWJ3J1ALHbB5/img-mY7eEWobILZ75quiRuxbJFWI.png?st=2024-08-31T18%3A15%3A54Z&se=2024-08-31T20%3A15%3A54Z&sp=r&sv=2024-08-04&sr=b&rscd=inline&rsct=image/png&skoid=d505667d-d6c1-4a0a-bac7-5c84a87759f8&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-08-30T23%3A39%3A04Z&ske=2024-08-31T23%3A39%3A04Z&sks=b&skv=2024-08-04&sig=WfH%2BBC4/zpvKI3NhTzDRt3DdGqXDEwyZWr%2Bvqnm92f4%3D

Page 7 Text: Page 4 - Setting: Damp and fog-shrouded city square enveloped in an eerie silence.

Generating image for Page 8 with prompt:

The man of the crowd pauses momentarily, his back turned to the narrator, creating an unsettling image of a lone figure in a desolate urban landscape, reinforcing the story's ominous atmosphere.

https://oaidalleapiprodscus.blob.core.windows.net/private/org-JIQuPumgUyJ9ld7QC8n1tutN/user-o2bO6Ni1LRgEDWJ3J1ALHbB5/img-0YTi88RbrsNsUgyet1O1zvL9.png?st=2024-08-31T18%3A16%3A21Z&se=2024-08-31T20%3A16%3A21Z&sp=r&sv=2024-08-04&sr=b&rscd=inline&rsct=image/png&skoid=d505667d-d6c1-4a0a-bac7-5c84a87759f8&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-08-30T23%3A16%3A23Z&ske=2024-08-31T23%3A16%3A23Z&sks=b&skv=2024-08-04&sig=a9swoYt9vjxbmLdnbQKMaRTWVuAGLX5dlCy1XuTOZ6s%3D

Page 8 Text: The man of the crowd pauses momentarily, his back turned to the narrator, creating an unsettling image of a lone figure in a desolate urban landscape, reinforcing the story's ominous atmosphere.

Generating image for Page 9 with prompt:

Let me know if you need more pages or further details.

https://oaidalleapiprodscus.blob.core.windows.net/private/org-JIQuPumgUyJ9ld7QC8n1tutN/user-o2bO6Ni1LRgEDWJ3J1ALHbB5/img-yoGtOOAkKctwUA04xBFyRHW5.png?st=2024-08-31T18%3A16%3A45Z&se=2024-08-31T20%3A16%3A45Z&sp=r&sv=2024-08-04&sr=b&rscd=inline&rsct=image/png&skoid=d505667d-d6c1-4a0a-bac7-5c84a87759f8&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-08-30T23%3A28%3A41Z&ske=2024-08-31T23%3A28%3A41Z&sks=b&skv=2024-08-04&sig=HJ7iKZFU4rOFAc239dW8QdPjTNAjXX2fdtZsxpvc1Rw%3D

Page 9 Text: Let me know if you need more pages or further details.

7. Conclusion

This research is aimed to explore the potential of AI to create seamless and customizable visual representations of literary texts, specifically focusing on the works of Edgar Allan Poe. The project sought to bridge the gap between traditional literature and modern digital media by enhancing the reader's visualization of narratives through AI-generated imagery, with the significance being how readers interact with literature in this digital age.

The methodology involved several key steps, starting with data preprocessing, which included text normalization, tokenization, stopword removal and sentiment analysis. These steps were crucial for preparing the text for machine learning models, ensuring that features used were relevant and informative. Genre classification was then performed using NLP techniques, and a Random Forest classifier was employed to further refine the model's understanding of the text. Finally, advanced NLP and image generation models-GPT-3.5-Turbo and DALL-E were used to create detailed and consistent visual representations of the text.

The results demonstrated the ability that AI models could effectively generate images that aligned with the narrative and tone of the text, although there were challenges related to consistency across pages, and the handling of complex narrative elements. The Use of GPT-3.5 Turbo and DALL-E provided a strong foundation for generating these visuals, but issues like rate limits and the complexity of prompts, to maintain a narrative throughout images highlighted areas for improvement. The ability to create a system of prompts with setting, characters, theme, tone and other subjects, the accuracy increased, and with further fine tuning, obtaining more diverse data, and exploring alternative models that may better handle the nuances of literacy text.

Next steps in this research would involve experimenting with different machine learning models, such as more advanced versions of transformers or hybrid models that combine text processing with large image generation. Additionally, expanding the dataset to include a broader range of literary styles and genres could improve the models versatility. Finally, refining the preprocessing steps and exploring new techniques for maintaining visual consistency would be crucial for this mode of research. Another mode of research would be to get the current model out to people, and for them to review the visual experience while reading their favorite books with the model. This could help develop a more multifaceted perspective on what aspects need to be focused on in a future model and how well the model did objectively to an audience.

In conclusion, this project has demonstrated the potential of AI in enhancing the reading experience through visual representation, offering new opportunities for engagement, accessibility, and innovation in the literary space.

8. Acknowledgements

I would like to thank Alaisha Alexander for the significant help she gave in learning about the AI space and the Inspirit AI team. I would also like to thank my parents for giving me the opportunity to learn more in this space.

9. References

McSpadden, Kevin. “Science: You Now Have a Shorter Attention Span than a Goldfish.” *Time*, Time, 14 May 2015, time.com/3858309/attention-spans-goldfish/.

Kelly, Kevin. “What Ai-Generated Art Really Means for Human Creativity.” *Wired*, Conde Nast, 17 Nov. 2022, www.wired.com/story/picture-limitless-creativity-ai-image-generators/.

Kelly, Kevin. “What Ai-Generated Art Really Means for Human Creativity.” *Wired*, Conde Nast, 17 Nov. 2022, www.wired.com/story/picture-limitless-creativity-ai-image-generators/.

Kelly, Kevin. “What Ai-Generated Art Really Means for Human Creativity.” *Wired*, Conde Nast, 17 Nov. 2022, www.wired.com/story/picture-limitless-creativity-ai-image-generators/.

“Explore Scientific, Technical, and Medical Research on ScienceDirect.” *ScienceDirect.Com | Science, Health and Medical Journals, Full Text Articles and Books.*, www.sciencedirect.com/.Accessed 31 Aug. 2024.

Collab notebook for Final Model
https://github.com/IanRmacdonnell/Short-story-storage

Hugging face model
HuggingFaceImageDescriptionGenerationAttempt