PG Certification Program in AI/ML

Capstone Project

AI Based Generative QA System

Group: 01

Task 1: Email Subject line Generation

Task 2: Question Answering on AIML Queries

Guide	Prof. Manish Shrivastava
Mentor(s)	Sagar Joshi Anubhav Sharma
Team	Yasoda Sreeram Kalluri Shaikh Iqbal Sikandar V K Rupesh Telaprolu

Code Repository	https://github.com/trup35/Capstone_Grp1
Cloud Deployment	https://huggingface.co/spaces/SilentLearner/AIML_Gradio

Contents

Abstract

At its core, this project aims to enhance Artificial Intelligence-driven text generation systems by focusing on the exploration of two critical tasks.: email subject line generation and domain-specific question answering in the field of Artificial Intelligence and Machine Learning (AIML). For the email subject line generation task, we address the challenge of creating concise and informative subject lines that effectively convey the content of emails. Leveraging the annotated Enron Subject Line Corpus, we develop a transformer decoder model based on the GPT architecture to autonomously generate subject lines from email bodies. The efficacy of our approach is evaluated using a variety of automatic metrics, gauging their alignment with human judgments. In the context of question answering on AIML queries, we tackle the problem of generating accurate answers to domain-specific questions. By adopting generative Question Answering (QA) systems, powered by efficient transformer decoder, specifically GPT, we aim to comprehend user intent and provide contextually relevant answers. Our investigation involves assessing the quality of generated answers through automatic metrics, establishing their correlation with human evaluations. This project contributes to the advancement of AI text generation by tailoring models to excel in real-world applications, such as enhancing email communication and providing targeted educational support. Through comprehensive experimentation and evaluation, we aim to push the boundaries of transformer-based text generation, ultimately refining its practical utility.

Introduction

The realm of artificial intelligence has witnessed significant breakthroughs with the emergence of pioneering tools like ChatGPT and DALL-E, marking the advent of a new AI era. In sharp contrast to conventional AI, which predominantly focuses on tasks such as classification and regression for analyzing existing data, these innovative technologies fall under the category of generative AI. Generative AI stands out by its unique capacity to craft novel content, spanning diverse domains like text and images. This creative approach, however, necessitates an initial understanding of existing data, as exemplified by text instructions, before generating fresh content. A noteworthy application of generative AI lies in its ability to generate high-dimensional data, serving as synthetic data to address data scarcity concerns in deep learning. This report delves into the distinctive characteristics of generative AI and its potential to reshape the landscape of artificial intelligence.

The primary emphasis of this study is on text generation, a specialized domain within Natural Language Processing (NLP) that merges computational linguistics and artificial intelligence to produce original textual content. This process involves creating syntactically and semantically accurate synthetic text. The process entails training a model that processes input data, comprehends contextual cues from the input, and subsequently generates new text aligned with the input data's domain. The generated text is expected to adhere to fundamental language structures while effectively conveying the intended message. The task of generating and assessing text for grammatical, semantic, and synthetic correctness is challenging due to the open-ended nature of text generation and its subsequent evaluation.

Recent strides in computational capabilities, alongside advancements in deep learning methodologies, have ushered in the era of automated text generation. Within this context, deep learning has made substantial contributions to diverse facets of natural language generation across a spectrum of tasks. These tasks encompass intricate endeavors such as dataset balancing, predictive modeling for subsequent words, text suggestions during conversations, crafting responses for question-answering systems, enabling chatbot interactions, facilitating machine translation, summarizing text content, classifying textual information, generating content for topic modeling, fostering dialogue creation, conducting sentiment analysis, composing poetry, scripting for cinematic narratives, and a multitude of other applications. These undertakings present formidable challenges within the realm of Natural Language Processing (NLP). Prior to the advent of pre-trained models (PTMs), addressing these challenges was intricate, as methods of that era grappled with limitations stemming from inadequate annotated data and constrained model parameters. Consequently, achieving fluency, coherence, and information richness in the generated text remained elusive. However, the emergence of large-scale PTMs has ushered in a transformative phase, wherein these models autonomously learn word combinations and sentence structures from unannotated data. This leap in sophistication notably enhances the prowess of these models in generating text that exudes fluency, coherence, and informative attributes.

Numerous pre-trained models are extensively employed in scholarly works to execute automatic text generation tasks. Among these models, GPT-2 stands out as a notable example, having been introduced by Redford *et.al.* GPT-2, characterized by its transformer-based architecture with an impressive 1.5 billion parameters, has undergone training on a substantial corpus of 40GB internet text extracted from a staggering eight million web pages. This transformative model is revered within the domain of text processing for its revolutionary nature. Particularly remarkable is its uncanny human-like proficiency in generating extensive sequences of text.

Email stands as a pervasive medium for digital communication, with an email message comprising two fundamental components: an email subject line and an email body. The subject line, visible to the recipient within their inbox, serves the purpose of succinctly conveying the essence of the email's content and the sender's intended message. The significance of a well-crafted subject line is underscored by its role in facilitating the efficient management of a considerable volume of emails

Within this project, Task 1 revolves around Subject Line Generation (SLG): an automated process of crafting email subjects based on the content of the email body. Although akin to email summarisation, these two tasks are distinct in their roles within the email composition and consumption process. While a subject line aids the sender in encapsulating the email's essence, a summary proves more valuable for comprehending lengthy emails from the recipient's perspective. The potential applications of automatically generated email subjects extend beyond initial email creation, encompassing downstream functions like email triaging to enhance email management efficiency. Additionally, in comparison to news headline generation or summarisation of single news documents, email subjects typically exhibit greater brevity. This necessitates a system capable of achieving high compression ratios while summarizing. As a result, we contend that this task can offer benefits to other forms of highly abstractive

summarisation, such as generating section titles for lengthy documents, thereby enhancing reading comprehension speed and accuracy

Conventional question-answering (QA) systems have assumed a pivotal role in catering to user inquiries. Nevertheless, these systems often necessitate manual design of user intents, potentially constraining their ability to adeptly handle intricate and diverse questions. As a result, there has been a burgeoning interest in generative question-answering systems, which exhibit the capability to produce responses that are more natural and coherent. Achieving this feat involves a deeper comprehension of context and user queries, enabling a more effective interaction.

In this study, our focus centers on the creation of a generative question-answering (GQA) system tailored for queries within the AIML domain. The system's primary objective is to comprehend participant questions and provide adept responses through a chatbot, leveraging the contextual cues embedded within the dataset. The development of such a system, however, presents a formidable challenge owing to the dearth of available data in the AIML domain. Notably, there is a conspicuous absence of question-and-answer datasets pertinent to AIML. To address this gap, a dataset has been meticulously curated through the collaborative efforts of seven capstone project teams. Each batch has meticulously annotated approximately 120 questions, each accompanied by two reasonably high-quality answers. The cumulative dataset encompasses a total of 1316 questions and answers earmarked for training purposes, along with 80 pairs for validation and an additional 120 pairs designated for testing purposes.

Problem Statement

This project is centered around the exploration of generative AI systems and encapsulates two distinctive objectives within this domain. The primary aim is to generate concise email subjects, utilizing a meticulously curated dataset, achieved through the fine-tuning of a GPT model to attain optimal outcomes. This endeavor involves acquiring a comprehensive grasp of the intricacies associated with implementing GPT model fine-tuning for subject line generation. Concurrently, during the process of fine-tuning the GPT model for subject line generation, a novel dataset is collaboratively constructed by all team members engaged in the capstone project. The secondary goal entails the training of a specialized Question-Answering (QA) model, intended for subsequent deployment to assess its proficiency in effectively addressing novel AIML queries.

1. Email Subject Line Generation

Unlike commonly tackled issues in fields like news summarization or headline generation, which possess conceptual resemblances to this undertaking, the distinctive challenge in task 1 revolves around crafting succinct and precise subject lines for emails. This endeavor necessitates pinpointing the most relevant sentences within email content and distilling the essence of the message into a concise selection of words. From an operational perspective, this project serves as a testing ground for NLP generative models, particularly various GPT-2 iterations. Furthermore, this project delves into evaluating text generation employing an array of metrics.

2. Question Answering on AIML Queries

Building upon the acquired knowledge of model finetuning and assessment from the initial task, the focal point of this task is centered around accomplishing the core objective of the second task: developing a domain-specific variant of the GPT model capable of addressing queries specific to the AIML course. While pretrained models exhibit proficiency in generating relevant text outputs for general and open-ended textual prompts, their capacity to produce refined outputs within specialized domains is limited. To address this, the conventional approach involves refining the model using a task-specific dataset to enhance its domain expertise. In this context, participants will collaborate to construct an innovative and contextually pertinent dataset tailored to the task at hand. Subsequent to the finetuning process, the performance of the model will be evaluated in relation to novel and related inquiries.

Literature review

Many articles have been published in the area of Generative AI, in particular Natural language generation and pre-trained language models. Out of which these articles are found relevant to the present project.

Rui Zhang and Joel R. Tetreault in 2019, studied the task of *email subject line generation by* automatically generating an email subject line from the email body. Authors developed a novel deep learning method and compared it to several baseline as well as state-of-the-art text summarisation systems. They also investigated the efficacy of several automatic metrics based on correlations with human judgments and proposed a new automatic evaluation metric.

Haifeng Wang *et al*., 2023, reviewed many articles published on the Natural Language Generation(NLG) tasks and they found that pre-trained models played a key role in improving the performance of NLG tasks. Further they emphasized that Large-scale pre-trained models automatically learn word combinations and sentence expressions from unannotated data, which significantly improves the models' ability in language generation in terms of fluency, coherence, and informativeness.

Bojic, I. et al., 2023, outlines the creation of a domain-specific extractive QA system employed as the AI component within a Human-AI health coaching model. Authors fine-tuned various domain-specific BERT models on their assembled dataset called SleepQA and conducted a comprehensive assessment of the resulting end-to-end extractive QA system using both automatic and human evaluation methods.

Tao Xiang 2023, created a generative question-answering(GQA) chatbot tailored for HR departments, its goal is to understand and effectively respond to employee questions by leveraging the context available in the dataset. Along with that, the author addressed a key challenge in developing such a chatbot i.e., the problem of lengthy input texts in user questions

and contexts. Author solved the issue by exploring the use of an efficient transformer, LongT5, and conventional transformer, T5 on the constructed three distinct datasets.

Methodology

The solution methodology employed for the text generation tasks encompasses a structured approach that integrates various components to facilitate the generation of coherent and contextually relevant text. The methodology is designed to address the unique requirements of each task while adhering to best practices in natural language processing and machine learning. The key elements of the solution methodology are outlined below:

Datasets Description

1. **The Annotated Enron Subject Line Corpus**

   The dataset designated as the Annotated Enron Subject Line Corpus, available at https://github.com/ryanzhumich/AESLC, has been earmarked for utilization in the inaugural task. It encompasses a curated subset of meticulously cleaned, filtered, and non-duplicated emails extracted from the comprehensive Enron Email Corpus, housing communication within the email inboxes of Enron Corporation employees.

   Notably, the evaluation split of this dataset, encompassing both development and test subsets, features three subject lines meticulously annotated by human evaluators. This provision of multiple potential references enhances the efficacy of evaluating the generated subject lines. This approach acknowledges the challenge inherent in identifying a single, distinct, and fitting subject line for each email.

   Furthermore, several dataset statistics provide valuable insights:

   1. Train / dev / test split sizes: 14,436 / 1,960 / 1,906 emails respectively.
   2. On average, an email contains 75 words.
   3. On average, a subject line comprises 4 words.

2. **AIML QA Corpus**

   The dataset presented here is the result of a meticulous collaborative effort involving seven distinct capstone project teams. Each of these teams has painstakingly annotated an

approximate total of 120 questions, each complemented by two answers of notably high quality. The annotation process adhered to the following guidelines:

- Questions were limited to 23 words or 150 characters.
- Answers were constrained to 35 words or 230 characters.

Slight flexibility was allowed within these limits to accommodate the intricacies of question-answer pairs. In aggregate, the compiled dataset comprises a grand total of 1316 question-answer pairs earmarked for training purposes. Additionally, 80 pairs have been set aside for validation, while a further 120 pairs have been reserved for rigorous testing.

Exploratory Data Analysis

Illustrating the contents of a text document visually stands as a pivotal task within the realm of text mining. For data scientists and specialists in Natural Language Processing (NLP), the exploration of document contents traverses diverse dimensions and levels of granularity. Not only do we delve into document content from various perspectives, but we also undertake the task of summarizing individual documents, revealing prevalent words and topics, identifying events, and crafting narratives.

Nevertheless, a disparity persists between the visualization of unstructured (text) data and structured data. For instance, several text visualizations do not directly represent the text itself; rather, they portray outcomes generated by language models, such as word count, character length, and word sequences.

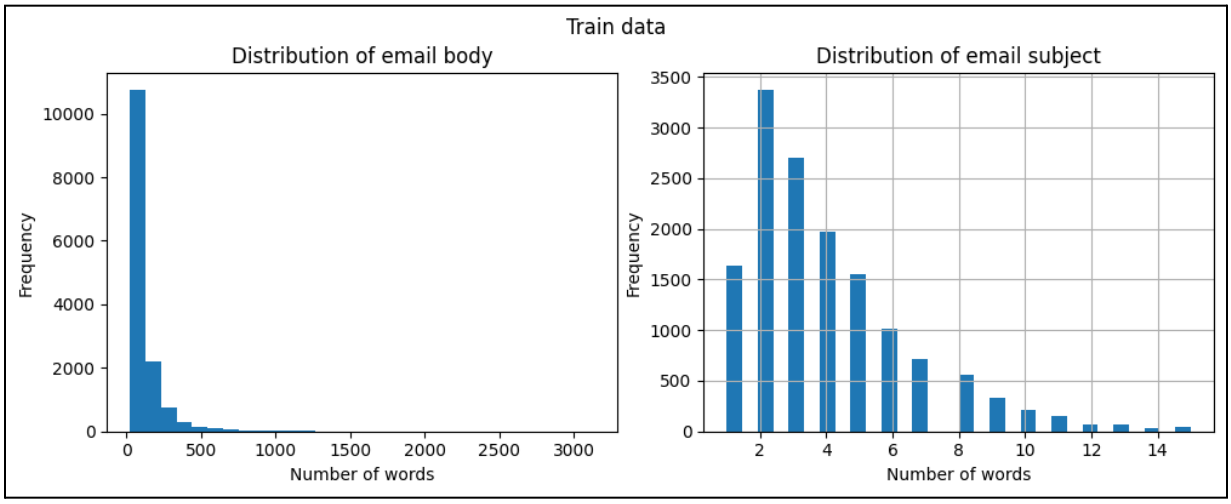

Figure 1: Distribution of email body and subject lengths of train data

Textual statistical visualizations encompass straightforward yet profoundly enlightening techniques, comprising the likes of word frequency analysis, assessment of sentence length, and evaluation of average word length. These methods prove invaluable for probing into the

foundational attributes of textual data. In pursuit of this exploration, our toolkit predominantly features histograms for continuous data and bar charts for categorical data.

Among the trio of extensively employed analyses, sentence length analysis emerges as a particularly enlightening endeavor in our present work. By subjecting this analysis to scrutiny, we unveil the distribution and mean of sentence lengths within both the body and subject sections of emails.

The following illustration displays the distribution patterns for both email body and subject lengths. As depicted in Figure 1, it is evident that the majority of email body lengths fall within the range of less than 500 words. Notably, a significant proportion of email subjects possess a length of 2 words; however, relying solely on this count is not advisable. The length of an email subject is intrinsically linked to the email's context.

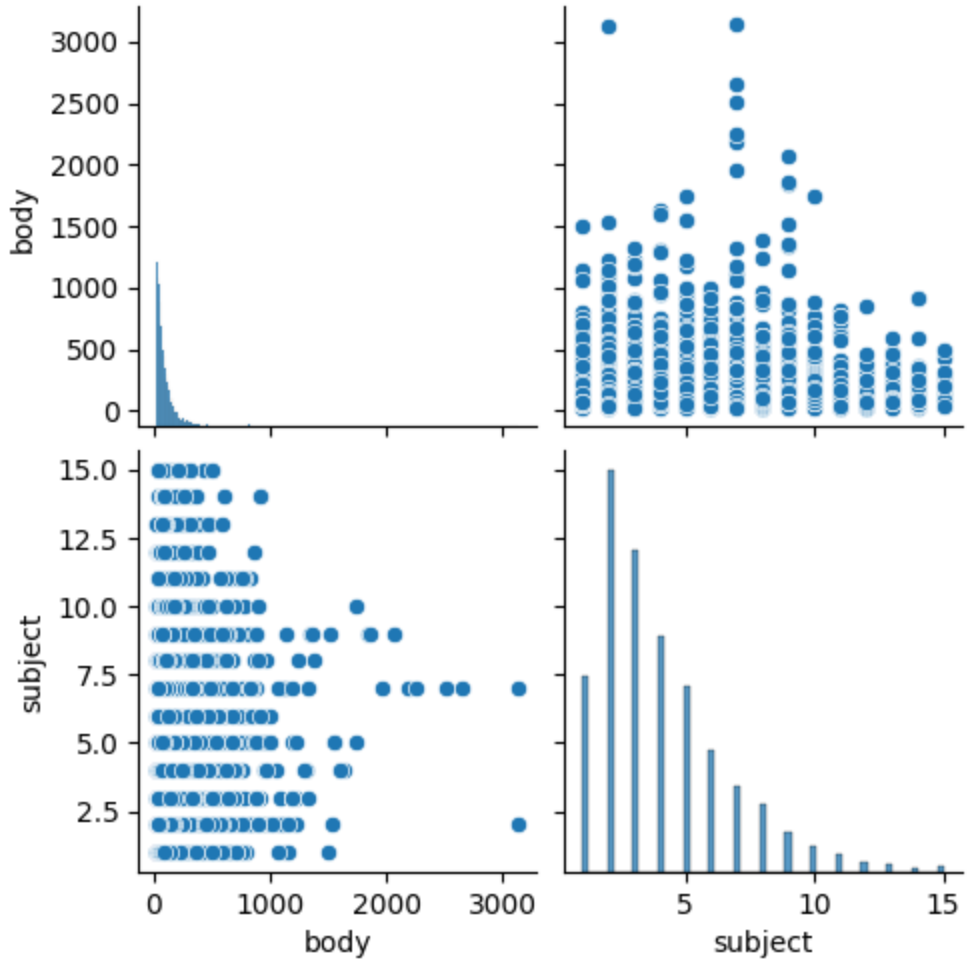

Figure 2: Pairwise correlation between body and subject lengths of train data

In terms of descriptive statistics for the training dataset, the mean lengths for the body and subject segments are recorded as 118 and 4 words, respectively. The standard deviation relative

to the mean for email body and subject lengths is calculated as 148.96 and 2.55, correspondingly. Noteworthy minimum and maximum lengths are noted for email bodies: 25 and 3136 words, and for email subjects: 1 and 15 words. Additionally, the median values for email body and subject lengths are found to be 74 and 3 words, respectively.

To delve into the potential correlation between email body and subject lengths, a pairwise correlation plot is introduced for sentence length analysis. This plot serves as a means to extract valuable insights regarding relationships between sentence lengths and other variables within the dataset. Figure 2 showcases the outcome of the pairwise correlation analysis for email body and subject lengths. The observation from this figure indicates an absence of correlation between subject and body lengths of emails, thus leading us to conclude that both lengths remain independent.

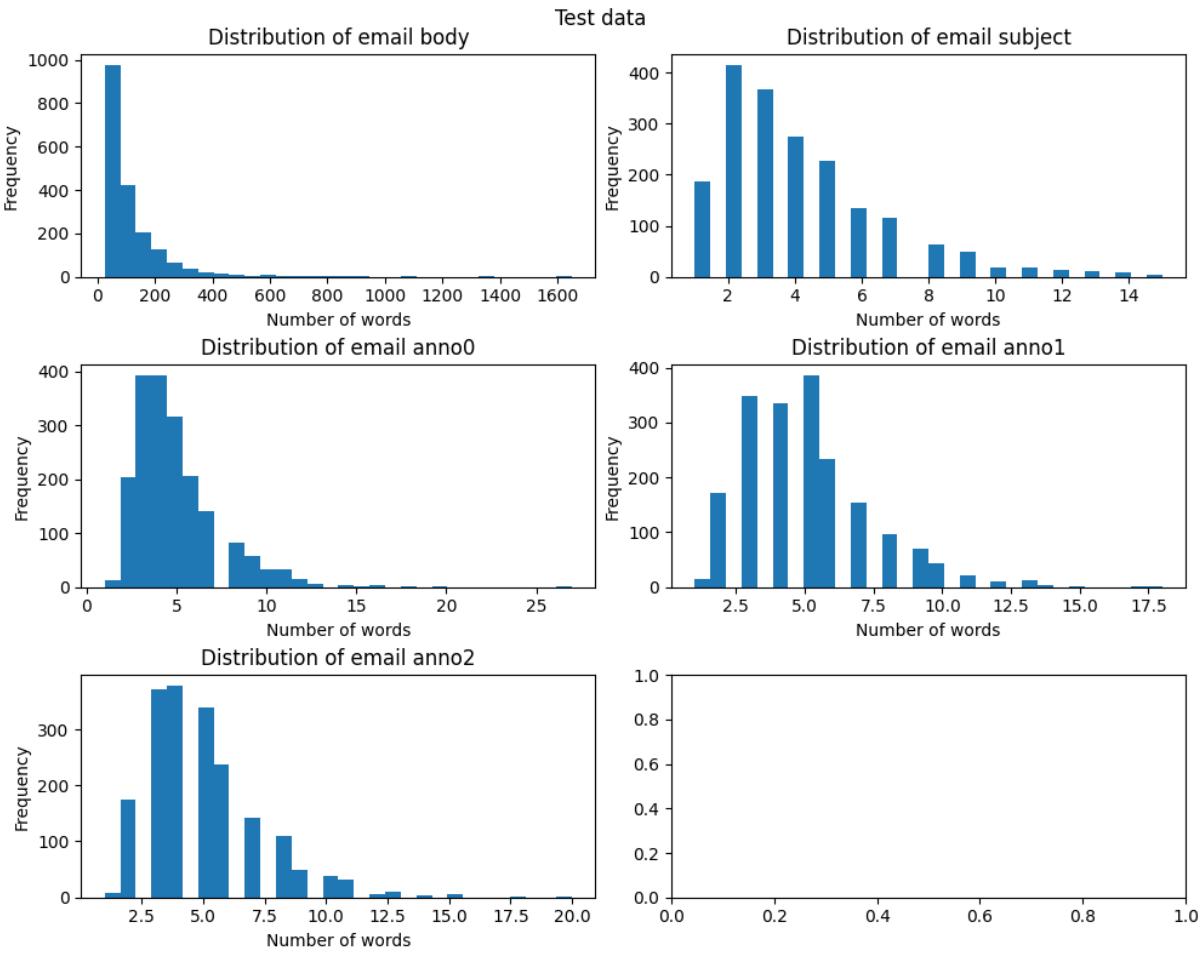

Figure 3: Distribution of email body and subject lengths of test data

Further insights into the distribution of email body and subject lengths for test dataset is visually presented in the subsequent figure.

Upon examining the distribution of email body and subject lengths within the test dataset, a discerning observation emerges from Figure 3. Notably, it becomes evident that the majority of email body lengths within the validation data fall within the range of fewer than 300 words, a range that is narrower compared to the body length distribution observed in both the train and validation datasets. Additionally, a considerable portion of email subjects, along with their accompanying annotations, exhibit lengths spanning from 2 to 5 words. However, it is crucial to exercise caution in interpreting these numbers, given that the length of an email subject is intricately linked to the specific contextual intricacies of the email.

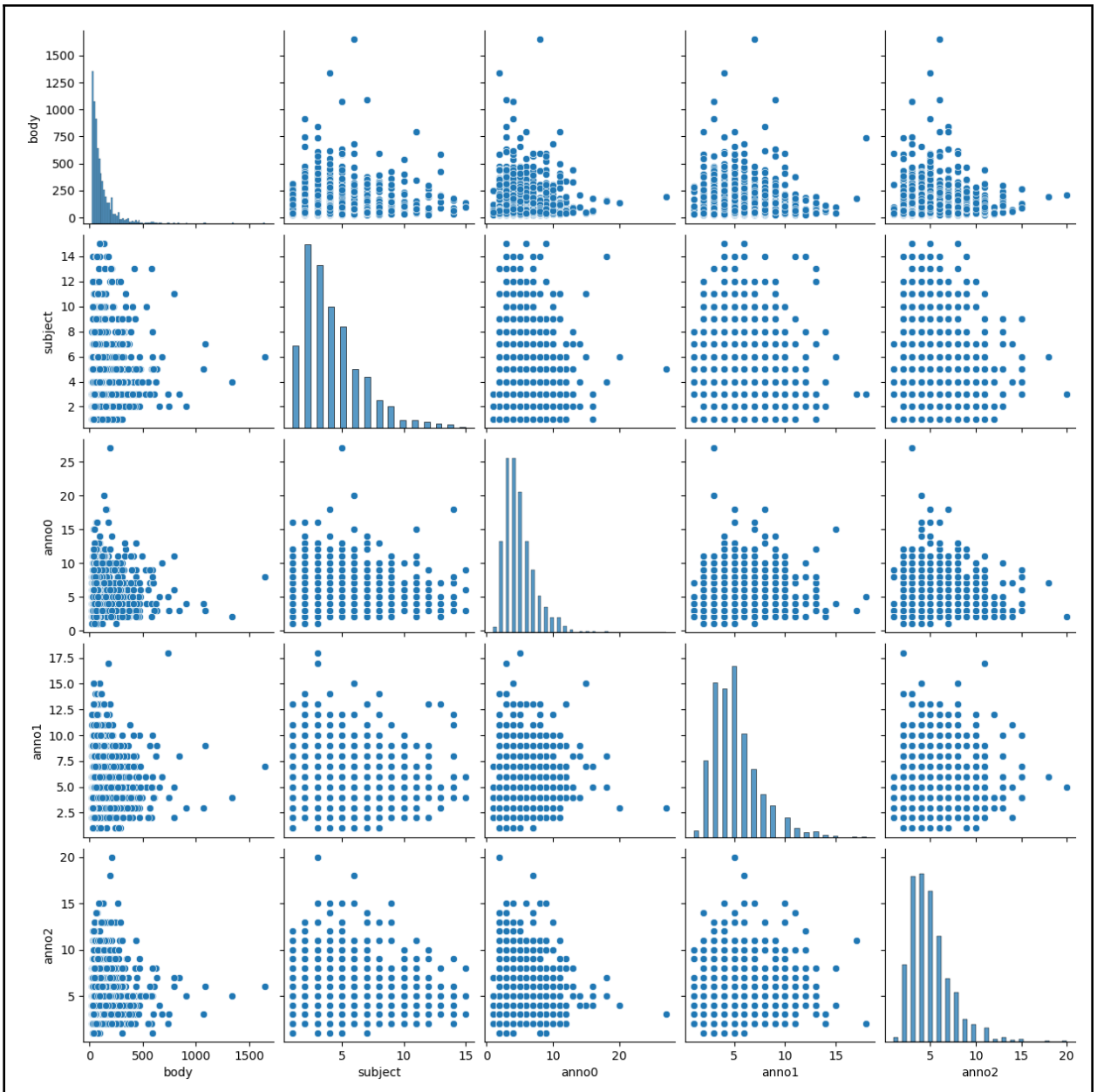

Figure 4: Pairwise correlation between body and subject lengths of test data

Directing our attention to the descriptive statistics of the test data, it is ascertained that the mean lengths for both the body and subject (inclusive of their annotations) are recorded as 114 and 5 words, respectively. The standard deviation, relative to the mean, for email body and subject lengths equates to 116.76 and 2.5, respectively. Moreover, the range of minimum and maximum lengths for email bodies extends from 25 to 1651 words, while for email subjects, the range spans from 1 to 20 words. Further insights into the central tendencies of the data are gleaned through median values, with email body and subject lengths registering at 78 and 5 words, respectively.

To illuminate potential correlations between email body and subject lengths, we have established a tailored pairwise correlation plot designed for sentence length analysis. This analytical tool serves as an avenue for extracting invaluable insights into the nuanced relationships that exist between sentence lengths and other variables housed within the dataset. The visual representation of this endeavor is encapsulated in Figure 6, where the pairwise correlation analysis of email body and subject lengths is vividly portrayed.

Upon meticulous scrutiny of Figure 4, it becomes apparent that no meaningful correlation exists between the lengths of subjects (including annotations) and the lengths of email bodies. This leads us to a resolute conclusion that all the lengths under consideration uphold their inherent independence.

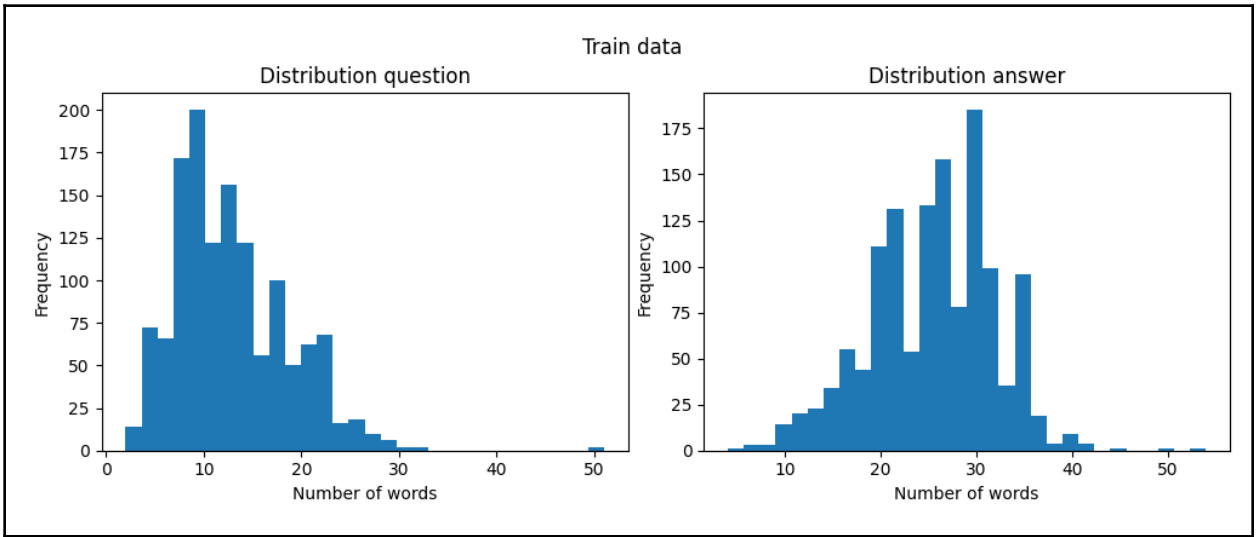

Figure 5: Distribution of questions and answers in train dataset of train data

In the context of Task 2, among the array of feasible exploratory data analysis techniques, sentence length analysis emerges as particularly enlightening. This method enables a nuanced understanding of the distribution of sentence lengths within the AIML QA dataset, shedding light on the characteristics of both questions and answers.

To unveil the underlying patterns, we've conducted sentence length analysis and translated the findings into visually informative representations. The resulting figures succinctly illustrate the distributions of questions and answers within the AIML QA dataset. These figures serve as

visual aids that encapsulate the disparities and trends in sentence lengths, offering valuable insights into the nature of the dataset's content.

This approach aligns with our objective of comprehensively understanding the dataset's structure and content, and subsequently leveraging these insights to refine our text generation strategies for Task 2.
Figure 5 visually unveils a distinct pattern, showcasing a prevalent concentration of questions around the 10-word mark and answers gravitating towards the 30-word range. Upon delving into the descriptive statistics of the training dataset, a comprehensive picture emerges:

For questions, the mean length stands at 12 words, accompanied by a standard deviation of 5.83, relative to the mean. The range spans from a minimum of 2 words to a maximum of 51 words. The median length mirrors the mean, registering at 12 words.

Shifting focus to answers, the mean length extends to 25 words, accompanied by a standard deviation of 6.45 relative to the mean. The range of answer lengths fluctuates between 4 and 54 words. The median, an indicator of central tendency, aligns closely with the mean at 26 words.

Figure 6 displays the results of the exploratory data analysis performed on the test data. This visualization unveils a distinct pattern, accentuating the predominant clustering of questions within the vicinity of the 10-word range. Notably, the distribution of answers also captures attention, with lengths of 1 and 2 correspondingly aligning with 23 and 30 words. To deepen our insights, a comprehensive comprehension can be developed by delving deeper into the descriptive statistics of the training dataset

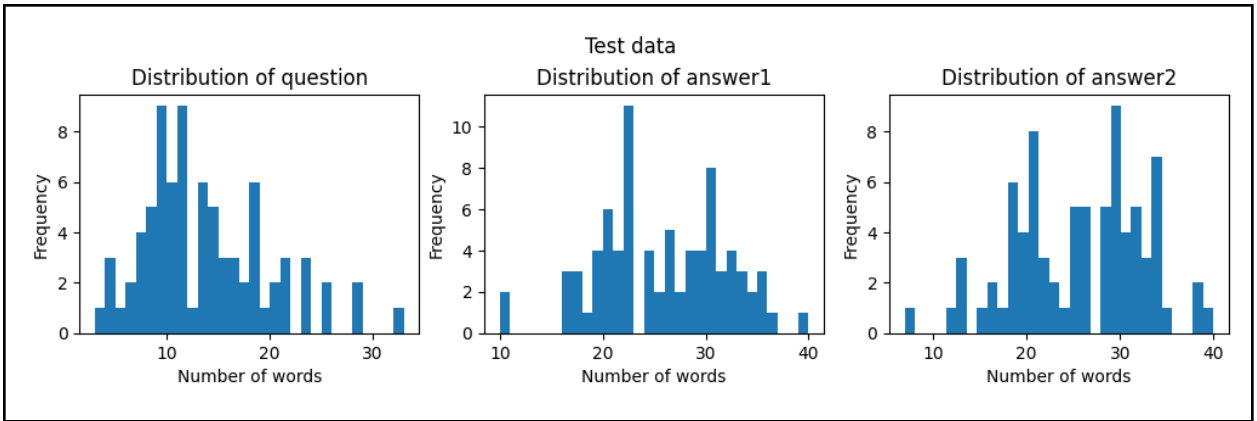

Figure 6: Distribution of questions and answers in test dataset of train data

Concerning questions, the average length stands at 13 words, partnered with a standard deviation of 6.34 in relation to the mean. The span of question lengths encompasses a minimum of 4 words and extends to a maximum of 41 words. Impressively, the median length closely mirrors the mean, registering at 12 words.

Shifting our attention to answers, the mean length extends to 26 words, accompanied by a standard deviation of 6.6 relative to the mean. The spectrum of answer lengths oscillates between 11 and 49 words. Intriguingly, the median—an indicator of central tendency—aligns closely with the mean at 26 words.

Tokenization

The Huggingface Transformers library serves as a Python toolkit designed to retrieve pre-trained models geared towards Natural Language Understanding tasks, encompassing functions such as sentiment analysis, and Natural Language Generation activities, including text generation and translation. Within its extensive repertoire, this library notably provides access to all four versions of GPT-2 that have been meticulously trained and published by OpenAI. This is augmented by an intuitive and user-friendly interface, rendering it exceptionally accessible for a wide range of users. Throughout the course of this discourse, we will frequently engage with three key concepts or classes intrinsic to the library:

1. **Tokenizer:** These components serve the purpose of storing the vocabulary specific to each model. Their utility is further extended through a collection of methods that enable the encoding and decoding of strings into a list of token embedding indices, which in turn serve as input for the model.
2. **Configuration:** These entities encapsulate the essential parameters requisite for constructing a model. Notably, they are not obligatory when working with a pre-trained model, thus streamlining the process.
3. **Model:** Representing PyTorch or Keras models, this class facilitates interaction with the pre-trained models featured in the library. It provides a robust platform for conducting tasks within the purview of these models.

Through the strategic deployment of these classes, the Huggingface Transformers library empowers users to seamlessly engage with a spectrum of Natural Language Processing endeavors, be it the analysis of text sentiment, the generation of textual content, or translation tasks.
Machine Learning models are not inherently designed to handle string inputs directly. Rather, a necessary preprocessing step involves tokenizing the input string. This process entails converting the string into a sequence of numerical values referred to as "tokens". These tokens subsequently undergo conversion into unique identifiers (IDs) using a lookup table. During model training or inference, it is these transformed tokens that are fed into the model.

For our GPT-2 model, we employ a tokenizer based on Byte-Pair Encoding subword segmentation. To initiate this process, the initial step entails loading the tokenizer through the interface provided by GPT2 classes within the Huggingface Transformers framework. Subsequently, the specific model version is designated, with OpenAI offering four dimensions for selection: 'gpt2', 'gpt2-medium', 'gpt2-large', and 'gpt2-xl'.

Functionally, the tokenizer serves three primary purposes:

1. It partitions the input text into tokens, which may not correspond precisely to individual words. Additionally, it performs the essential tasks of encoding and decoding these tokens into input IDs, and vice versa.
2. The tokenizer facilitates the incorporation of new tokens into the existing vocabulary.
3. It adeptly manages special tokens such as masks, text initiation and conclusion indicators, and specialized separators, among others.

To align the text with the model's anticipated format, we turn to the 'encode' and 'batch encoding' functions. These operations allow us to define the tensor format—whether PyTorch or TensorFlow. Due to the inherent variability in batched inputs' lengths, strategies such as padding and truncation come into play. Padding ensures uniform lengths by introducing a special padding token to shorter sequences. Conversely, truncation addresses lengthier sequences by curtailing their extent.

Typically, the approach of padding batches to match the length of the longest sequence and truncating to the model's maximum length suffices. However, the API accommodates additional strategies for customization. When engaging with the API, three pivotal arguments—'padding', 'truncation', and 'max_length'—enable the tailored management of text sequences.

GPT-2, as a model, incorporates absolute position embeddings to provide context and positional information to the tokens within a sequence. This aspect ensures that the model can effectively understand the sequential order of tokens, which is crucial for generating coherent and contextually accurate text.

To harness the benefits of these absolute position embeddings, it's advisable to pad the inputs on the right side of the sequence rather than the left. Padding on the right ensures that the original tokens retain their positions and relative orders, while the added padding tokens are positioned at the end of the sequence. This approach maintains the integrity of the context provided by the absolute position embeddings, preventing any disruption in the sequential understanding that the model relies upon.

Before embarking on the tokenization process, a well-conceived prompt needs to be devised to suit the model's input requirements. For Task 1, our objective entails generating an email subject based on the content of the email body. To effectively train the model, both the body and subject of the email must be introduced as inputs to the model in a cohesive manner. During inference, however, the model shall be provided solely with the email body, and it is anticipated to generate a subject followed by the body. Consequently, we define the following formats for training and inference prompts:

Training input sequence: '\<body>' Body of email '\<subject>' Subject of email.

Inference Prompt: '\<body>' Body of email '\<subject>'.

For Task 2, the goal revolves around generating answers to questions while utilizing domain-specific questions as inputs. During model training, it is essential to furnish the model with question and answer pairs as inputs. Inference, on the other hand, involves supplying the model with solely the question as input, anticipating the model to generate an answer followed by the question prompt. Thus, we establish the ensuing formats for training and inference prompts:

Training input sequence: '\<question>' Question '\<answer>' Answer.

Inference Prompt: '\<question>' Question '\<answer>'.

Subsequently, these prompts are translated into tokens and further integrated into the lookup table through the predefined tokenizer. This meticulous transformation process is fundamental in ensuring that the model seamlessly comprehends and interacts with the provided prompts, ultimately contributing to the efficacy of the entire workflow.

Model and Evaluation

Within the GPT-2 family, there are different model classes that serve specific purposes. Two of these frequently used classes are GPT2Model and GPT2LMHeadModel.

GPT2 Model is the foundational building block of the GPT-2 architecture. It is primarily used for tasks like language understanding, where the model processes input text to generate contextualized representations of the tokens. However, it does not include the head responsible for text generation. GPT2Model takes tokenized input text and generates embeddings that capture the contextual information of each token. These embeddings can then be used as features for downstream tasks like sentiment analysis, named entity recognition, and more. It's not directly used for text generation but for understanding and representing text.

The GPT2LMHeadModel, which stands for "Language Modeling Head," is specifically designed for text generation tasks. It includes the text generation head on top of the GPT-2 base architecture. GPT2LMHeadModel takes tokenized input text and generates coherent and contextually relevant text as output. It's used for tasks like creative writing, text completion, and chatbot responses. The autoregressive nature of the model allows it to predict the next token in a sequence based on the preceding tokens, leading to natural-sounding text generation.

GPT2Model primarily caters to language understanding tasks by offering contextual embeddings for text analysis, whereas the GPT2LMHeadModel is specifically designed for text generation tasks, allowing the coherent generation of text based on provided prompts. Given the objectives of this project, where the aim is to generate text in the form of subject lines and provide answers within the AIML domain, the GPT2LMHeadModel is the more suitable model class to employ.

Throughout this project, we utilize the GPT2LMHeadModel along with various pretrained variants of the GPT-2 architecture. These pretrained models have been fine-tuned to generate text that adheres to specific tasks, such as subject line generation and answering queries in the AIML domain. This choice empowers us to leverage the model's inherent text generation capabilities, enabling the production of coherent and contextually relevant text outputs. By employing different pretrained variants of the GPT-2 model, we can explore various nuances of text generation across different tasks, enhancing the overall effectiveness and versatility of our approach.

Evaluation is a pivotal phase in assessing the proficiency and caliber of text generation models. Within the scope of our project, we have adopted a diverse range of evaluation metrics to meticulously evaluate the performance of our text generation models. These metrics have been meticulously selected to provide a holistic view of the generated content, enabling us to ascertain the model's prowess in producing high-quality text.

Our evaluation framework leverages a synergy of automated metrics and human judgment, recognizing the intricate and subjective nature of language. This integrated approach ensures a robust evaluation that captures various dimensions of text quality. Below, we outline the principal evaluation metrics we've employed to comprehensively assess the quality of text generated by models such as GPT-2:

- **Automated Metrics:** These objective measures gauge the alignment of generated text with reference or ground truth text. They offer quantitative insights into the model's predictive and linguistic capabilities. Our chosen automated metrics include:
    - **Perplexity:** Quantifying the model's predictive power by measuring its token-level prediction performance.
    - **BLEU, ROUGE, METEOR:** Evaluating n-gram overlaps, semantic similarity, and grammaticality against reference text.
    - **BERTScore:** Measures semantic similarity by comparing embeddings of generated and reference text.
- **Human Judgment:** Incorporating the human perspective is indispensable for evaluating the nuanced aspects of text quality. We employ human assessors to rate generated text based on fluency, coherence, relevance, and overall quality. Their expert judgment offers a qualitative dimension to our evaluation.

BERTScore is a specific automated metric that leverages pre-trained BERT embeddings to calculate the semantic similarity between generated text and reference text. It goes beyond surface-level overlaps by considering contextual understanding and word meaning. BERTScore is particularly relevant for evaluating the quality of generated text in tasks like question answering, where capturing semantic nuances is crucial.

The term "automated metrics" refers to a broader category of quantitative evaluation measures used in natural language processing. "BERT metric" specifically refers to the utilization of BERT embeddings to measure semantic similarity between generated and reference text. BERTScore is a concrete example of an automated metric that falls within the category of "BERT metric" and is

applied to tasks like question answering. When comparing these terms, it's important to recognize that BERTScore is a specific type of automated metric that addresses certain limitations of traditional metrics by considering semantic alignment. Evaluation strategy is a well-balanced fusion of both automated and human-centric evaluation approaches. This comprehensive methodology provides a thorough and multi-faceted assessment of the text generation models we've employed, ensuring that their outputs meet the highest standards of quality and coherence.

Hyperparameter Optimization

Hyperparameter optimization plays a vital role in enhancing the performance and effectiveness of natural language generation (NLG) models. Optimizing hyperparameters involves fine-tuning the configuration settings of the model to achieve optimal results. Hyperparameter optimization for GPT-2 text generation tasks involves tuning the parameters that significantly affect the model's performance in generating coherent and contextually relevant text.
In our investigation of text generation tasks using GPT models, we have thoroughly examined the impact of various critical hyperparameters on the model's performance. Among the available options, we have specifically focused on the variability of the following hyperparameters:

Batch Size:

By systematically varying the batch size during training, we've explored its influence on training efficiency, convergence speed, and memory consumption. Different batch sizes impact the model's ability to generalize and the computational resources required.

Indeed, GPT-2's substantial size and the memory requirements it demands pose challenges, particularly when it comes to determining an optimal batch size for training. Given the constraints of available resources, such as the 15 GB GPU compute offered by Google Colaboratory, it's necessary to strike a balance between model performance and training time efficiency. This balance often leads to an exploration of feasible batch sizes, with consideration for both memory limitations and training duration.

In our analysis, we've specifically focused on batch sizes that are manageable within the confines of the available resources. The batch sizes of 4, 8, 16, and 32 have been chosen based on practical considerations, acknowledging that larger batch sizes could potentially exceed the GPU's memory capacity. Despite the trade-off between batch size and training time, we have been able to discern certain patterns in model performance as batch sizes are adjusted.

The observed trends underscore the intricate relationship between batch size, memory consumption, and training duration. While smaller batch sizes lead to prolonged training times, they offer the advantage of fitting within limited memory resources. Conversely, larger batch sizes may expedite training but can surpass memory limits, thereby impacting model stability and performance.

Our analysis serves to shed light on the interplay between these factors, equipping us with valuable insights for making informed decisions about batch size selection. By navigating these challenges and strategically choosing appropriate batch sizes, we are working to optimize both model performance and the training process within the available resources.

GPT-2 Variant:

We've delved into the effect of utilizing different GPT-2 variants, ranging from small to large, on text generation quality. This exploration helps us understand how gpt2 type influences the generation of coherent and contextually relevant text.

In our current study, we have diligently narrowed down the selection to five distinct GPT-2 models, each chosen based on a combination of factors such as popularity, download count, and recency. These selections were made by meticulously considering models specifically tailored for text generation tasks. However, it's worth noting that the initial pool of candidates encompassed more than ten models. Despite the initial pool, constraints imposed by the available GPU compute resources, limited to 15 GB in the case of Google Colaboratory, dictated the final selection.

The following GPT-2 models have been chosen for our analysis due to their compatibility with the 15 GB GPU memory constraint:

1. 'distilgpt2'
2. 'gpt2'
3. 'Ar4ikov/gpt2-650k-stable-diffusion-prompt-generator'
4. 'crumb/gpt2023'
5. 'olm/olm-gpt2-dec-2022'

The selected batch sizes have been intentionally calibrated to balance memory limitations with training efficiency and model performance. This thorough selection process ensures that our analysis is both comprehensive and practical, enabling us to derive meaningful insights from the chosen models and batch sizes. The combination of model compatibility and batch size optimization empowers us to perform a thorough examination of text generation capabilities while working within the constraints of available computational resources.

Learning Rate:

Our investigation involves altering the learning rate to understand its impact on model convergence and generalization. Optimizing the learning rate ensures effective weight updates during training.

In our experimentation phase, we meticulously varied the learning rates—0.00005, 0.0001, 0.0005, and 0.001—to comprehensively explore their impact on fine-tuning the GPT-2 model. This exploration aimed to pinpoint the optimal learning rate that strikes the right balance between achieving improved model performance and maintaining efficient training.

To further refine the learning process, we employed learning rate schedules, specifically leveraging techniques like linear warm-up. This strategy gradually adapts the learning rate, enabling a smoother transition between different stages of training. The outcomes of these experiments revealed a significant trend: as the learning rate decreased, the model's performance demonstrated enhancement.

However, it's important to highlight an accompanying observation: the reduction in learning rate correlated with a substantial increase in training time. This is an expected outcome, as lowering the learning rate extends the duration required for convergence.

These findings underscore the delicate equilibrium between model performance and training efficiency. As we continue to fine-tune the GPT-2 model, we are attentively considering the trade-offs between enhanced performance and the time investment necessitated by lower learning rates. This awareness empowers us to make strategic choices that align with the specific requirements and goals of our text generation tasks.

Max Sequence Length:

We've analyzed how varying the maximum sequence length affects the model's ability to handle different input contexts. Adjusting this hyperparameter can influence the generation of longer or shorter text segments. Through our comprehensive exploratory data analysis, we've gained valuable insights into the distribution of text lengths for the two tasks under consideration.

For **Task 1**, a detailed analysis of email bodies has revealed that a significant portion falls within the range of 0 to 500 words. In contrast, subject lengths are notably shorter, inherently influenced by the context of each email. As a result of this discrepancy, we've opted to determine the maximum sequence length by considering the email body length. For this task, the maximum sequence length ranges from 120 to 512 words, accommodating the variations in email body lengths and ensuring effective subject generation.

Transitioning to **Task 2**, our approach to annotating the dataset has led us to impose constraints on question and answer lengths—capping them at 25 and 35 words, respectively. This decision sets a natural boundary for the maximum sequence length, which doesn't exceed 60 words. However, recognizing the significance of experimentation, we've extended our exploration to encompass max length values of 60, 70, 80, and 90 words for Task 2. This broadens our understanding of how varying sequence lengths impact the quality and coherence of the generated answers.

By aligning our maximum sequence lengths with the intrinsic characteristics of each task's data and requirements, we are optimizing the text generation process to yield output that maintains relevance, coherence, and context. This thorough consideration of sequence lengths bolsters the quality of the generated content and aligns with the specific objectives of each task.

By meticulously examining these hyperparameters, we aim to gain insights into how they interact with the GPT model's architecture and influence text generation outcomes. Our focus on

these specific hyperparameters allows us to make informed decisions regarding their optimal values for our text generation tasks. This exploration empowers us to fine-tune the model's configuration, leading to improved performance, more efficient training, and better-textured generated output.

Table 1: Levels of hyperparameters considered for task 1

Sl. No	Batch size	Gpt2 type	Learning Rate	Max length
1	4	distilgpt2	**0.0001**	**256**
2	8	gpt2	0.0005	384
3	16	Ar4ikov/gpt2-650k-stable-diffusion-prompt-generator	0.001	448
4	32	crumb/gpt2023		512
5		**olm/olm-gpt2-dec-2022**		

To summarize, the subsequent tables outline the diverse ranges of hyperparameters that have been under scrutiny for the tasks within this project. Table 1, presented below, furnishes insights into the various tiers of hyperparameters explored for task 1, while Table 2 delves into the specifics of hyperparameters examined for task 2.

Table 2: Levels of hyperparameters considered for task 2

Sl. No	Batch size	Gpt2 type	Learning Rate	Max length
1	1	distilgpt2	0.00005	80
2	2	gpt2	0.0001	60
3	4	Ar4ikov/gpt2-650k-stable-diffusion-prompt-generator	0.0005	70
4	8	crumb/gpt2023	0.001	90
5	16	olm/olm-gpt2-dec-2022		
6	32			
7	64			

In the pursuit of hyperparameter optimization, a conversing strategy is employed, a technique frequently applied across diverse scientific challenges. This approach bears resemblance to the

concept of varying one parameter at a time, as commonly used in the realm of experimental design. The conversing strategy operates as follows: initially, the hyperparameters are ranked in order of perceived importance. Subsequently, a systematic process unfolds whereby the levels of the first hyperparameter are alternated while maintaining the default levels for the other hyperparameters. Once a consensus is reached regarding the optimal configuration of the first hyperparameter, attention turns to the second hyperparameter, which is iteratively adjusted while the first parameter remains at its optimized state, and other parameters remain at their defaults. This iterative procedure continues until saturation is achieved across all hyperparameters. This methodology ensures a thorough exploration of the hyperparameter space and aids in pinpointing an optimal configuration for model performance.

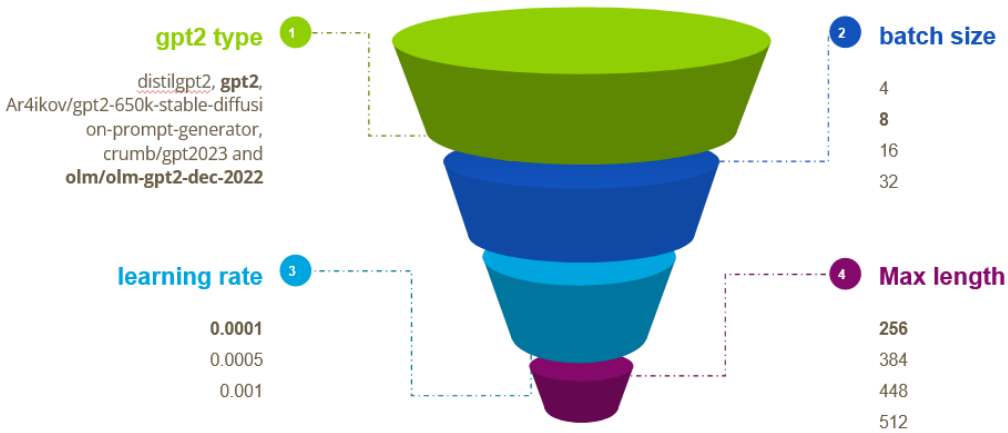

Figure 10: Hyperparameter optimization strategy for task 1

The ensuing figures, namely Figure 10 and Figure 11, visually exemplify the process of hyperparameter optimization for both task 1 and task 2. This optimization approach involves systematically ranking the hyperparameters and subsequently evaluating their effects across distinct levels within each hyperparameter. These figures offer a comprehensive depiction of how this ranking-based hyperparameter optimization strategy has been methodically implemented to enhance the performance of both task 1 and task 2.

The outcomes resulting from the implementation of the aforementioned hyperparameter optimization strategy are articulated and elaborated upon in the "Results and Discussion" section. This section serves as the platform where the achieved results are meticulously presented and analyzed, providing insights into the impact and effectiveness of the applied hyperparameter optimization approach.

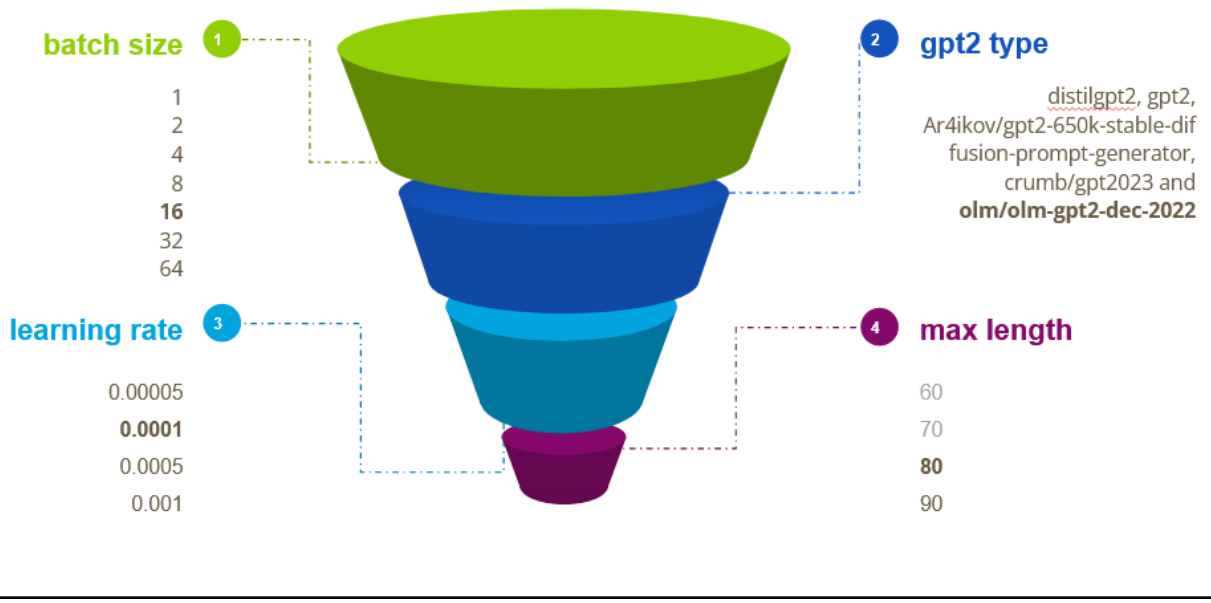

Figure 11: Hyperparameter optimization strategy for task 2

Results and Discussions

This section constitutes a pivotal segment of our analysis, offering a comprehensive exploration and evaluation of the outcomes derived from our text generation tasks. This section not only presents the empirical findings but also engages in an in-depth interpretation and scrutiny of the observed results. By juxtaposing the generated content with the anticipated outcomes, we aim to unravel the effectiveness and performance of our text generation models, specifically the GPT-2 based approaches employed.

Through the meticulous application of the converging strategy for hyperparameter optimization, as previously delineated, our project has yielded valuable insights and outcomes across the two tasks under investigation. The diverse array of hyperparameter configurations that were meticulously explored have provided us with a comprehensive perspective on the interplay between these parameters and the text generation processes we are addressing. These findings serve as a testament to the efficacy of the converging strategy in fine-tuning our text generation models for optimal performance. By juxtaposing empirical outcomes against the backdrop of varying hyperparameter configurations, we gain a nuanced understanding of how these choices impact the quality, coherence, and relevance of the generated text.

In the context of Task 1, which pertains to email subject line generation, our approach involved applying the converging strategy in conjunction with exploring boundary cases. This comprehensive methodology was employed to bolster the robustness and reliability of our results. Through these endeavors, we arrived at a configuration that has demonstrated commendable performance across the metrics we have deemed relevant.

The metrics chosen for this analysis, namely BLEU and ROUGE, have been meticulously evaluated to gauge the efficacy of our approach. It's noteworthy that these metrics provide complementary perspectives, further enhancing the comprehensiveness of our assessment. Despite their distinct formulations, both metrics have exhibited similar trends and patterns in response to our optimization efforts.

Given this congruence, we leverage the ROUGE-R1 and BLEU metrics to effectively illustrate the results obtained from our model. This strategic selection aligns with the consistency and convergence observed across these metrics, bolstering our confidence in the outcomes achieved through the converging strategy. By utilizing these metrics in tandem, we present a holistic depiction of the model's performance in generating email subject lines, enriching our understanding and enabling us to draw reliable conclusions from our endeavors.

To facilitate straightforward differentiation among the various configurations resulting from hyperparameter optimization, we adopt a distinctive notation system denoted by the acronym "BGLM," followed by numerical values. In this system, each letter signifies a specific hyperparameter, while the accompanying number designates the particular level chosen for that parameter. The breakdown is as follows:

- "B" signifies batch size,
- "G" represents the GPT-2 variant,
- "L" stands for learning rate, and
- "M" denotes the maximum sequence or context length.

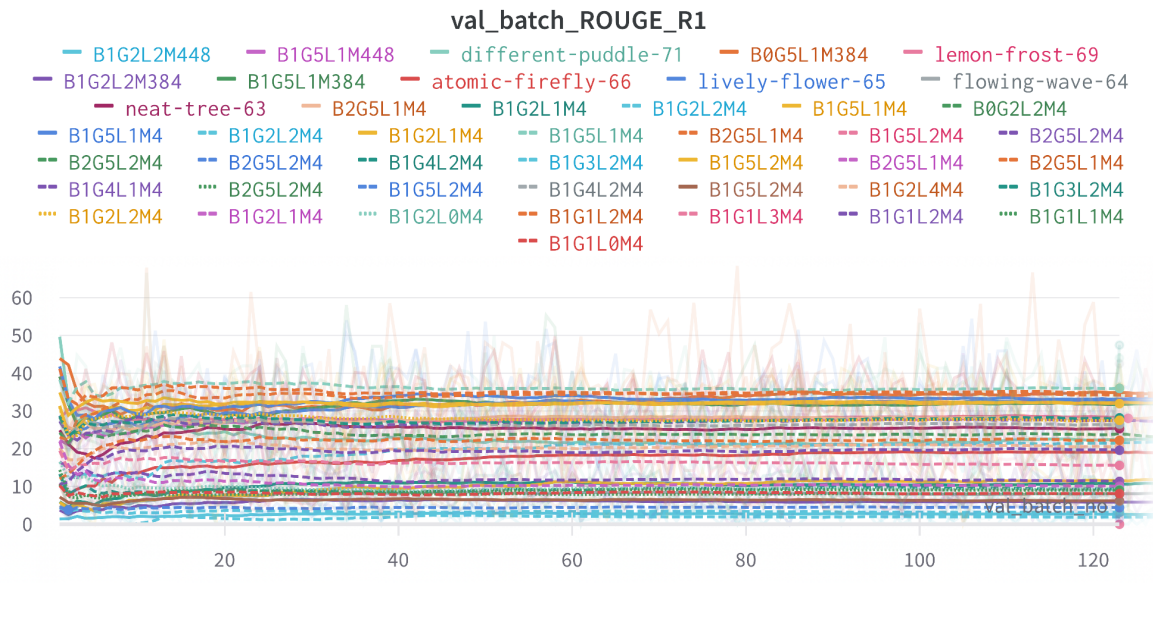

Figure12: ROUGE_R1 metric scores for all the configurations of task1

The numeric value following each letter encapsulates the specific level that has been considered for that corresponding hyperparameter. Through this concise yet informative notation, we establish a clear and easily recognizable framework that enables quick identification and comparison of diverse hyperparameter configurations across our optimization efforts.
Figures 12 and 13 distinctly illustrate a compelling trend across various configurations that were systematically devised through hyperparameter optimization. Notably, the ROGUE-R1 score and BLEU metric for a specific configuration, namely "B1G5L1M4," consistently exhibit notable magnitudes across the entirety of validation batches.

In detail, the "B1G5L1M4" configuration embodies the following parameter selections:

- Batch size of 8
- GPT-2 type: 'olm/olm gpt2 dec 2022'
- Learning rate: 0.0001
- Maximum sequence/context length: 256

This configuration has consistently demonstrated remarkable performance, as evidenced by the substantial ROUGE-R1 and BLEU scores. Such an observation reinforces the effectiveness of the chosen hyperparameters within the context of the validation dataset, substantiating their positive impact on the quality and coherence of the generated email subject lines.

The persistent excellence showcased by the "B1G5L1M4" configuration across the validation batches underscores its robustness and suitability for email subject line generation, thus contributing to the overarching objectives of our text generation task.

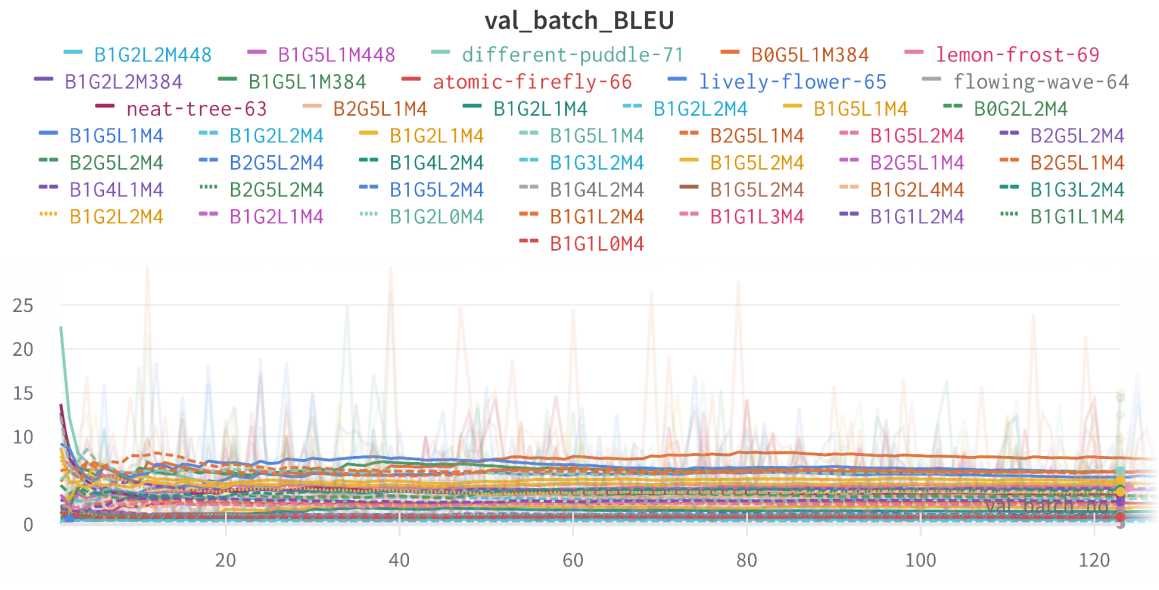

Figure13: BLEU metric scores for all the configurations of task1

Table 3 displays the BLEU and ROUGE metrics for the best-performing configurations of the 'olm/olm gpt2 december 2022' model on the test data. The two highlighted configurations demonstrate strong performance during deployment, providing valuable insights into model effectiveness through their respective BLEU and ROUGE scores.

Table 3: BLEU and ROUGE metric scores for task 1 best-performing configurations

Model	BLEU	ROUGE-R1	ROUGE-R2	ROUGE-RL	ROUGE-RLsum
B1G5L1M4	6.03	35.98	19.34	34.08	34.02
B2G5L1M4	6.03	34.22	17.55	32.20	32.18

In the context of Task 2, focused on question and answering for AIML queries, our strategy involved implementing the converging strategy alongside the exploration of boundary cases. This comprehensive methodology aimed to fortify the reliability and resilience of our results. Through these combined efforts, we identified a configuration that has consistently demonstrated commendable performance across the relevant metrics we have considered.

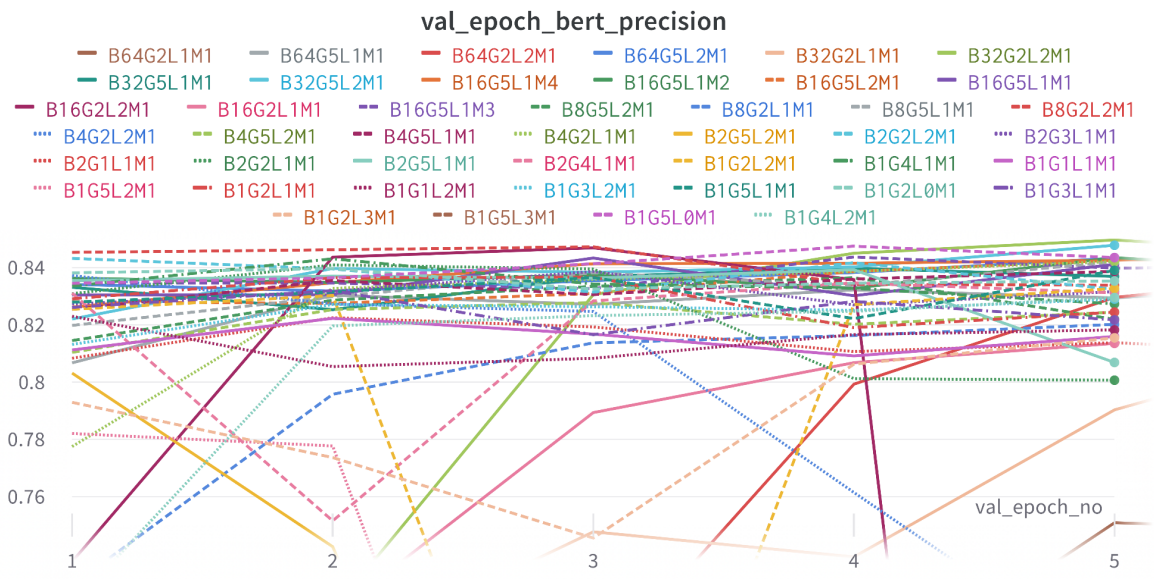

Figure14: BERT precision scores for all the configurations of task 2

The selected metrics for this analysis, including BLEU, ROUGE, and BERT metrics, were meticulously evaluated to assess the effectiveness of our approach. Notably, these metrics offer complementary perspectives, enhancing the overall comprehensiveness of our assessment. Despite their unique formulations, all metrics have displayed similar trends and patterns in response to our optimization endeavors.

Building upon this congruence, we highlight the BERT metric to effectively showcase the results obtained from our model. This strategic choice aligns with the consistent and converging patterns observed across these metrics, bolstering our confidence in the outcomes achieved through the converging strategy. By synergistically utilizing these metrics, we present a holistic representation of the model's performance in generating answers for AIML queries. This approach enriches our understanding and enables us to draw reliable conclusions from our efforts.

For the purpose of distinguishing among various configurations resulting from hyperparameter optimization, we adopt a distinctive notation system labeled as "BGLM," followed by numerical values. This system employs letters to represent specific hyperparameters, and the accompanying numbers signify the corresponding level chosen for each parameter. Specifically:

- "B" represents batch size,
- "G" denotes the GPT-2 variant,
- "L" signifies the learning rate, and
- "M" designates the maximum sequence or context length.

To differentiate from the previous problem, actual batch size values (1, 2, 4, 8, 16, 32, and 64) are used instead of numerical levels for batch size. This concise yet informative notation streamlines the identification and comparison of diverse hyperparameter configurations within our optimization efforts.

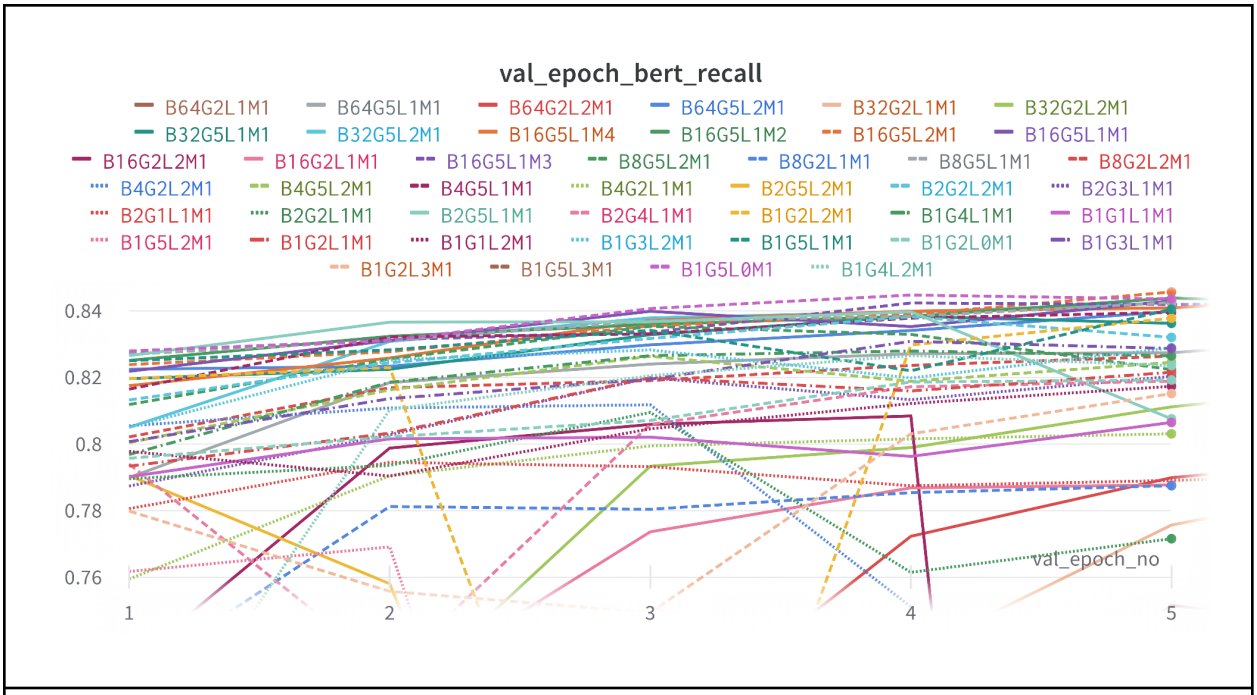

Figure15: BERT recall scores for all the configurations of task2

Figures 14, 15, and 16 visually elucidate a compelling trend observed across multiple configurations resulting from hyperparameter optimization. Notably, the BERT precision, recall,

and f1 metrics demonstrate a consistent proximity to each other. This indicates that determining a clear winner, as in the previous case, is not straightforward. Consequently, human evaluation was employed to determine the best-performing configuration. Models with BERT scores approximating 0.84 were chosen, and inference was conducted with randomly selected questions not present in the training data. Based on these inferences, it was determined that the "B16G5L1M1" configuration yielded favorable results with the random questions considered.

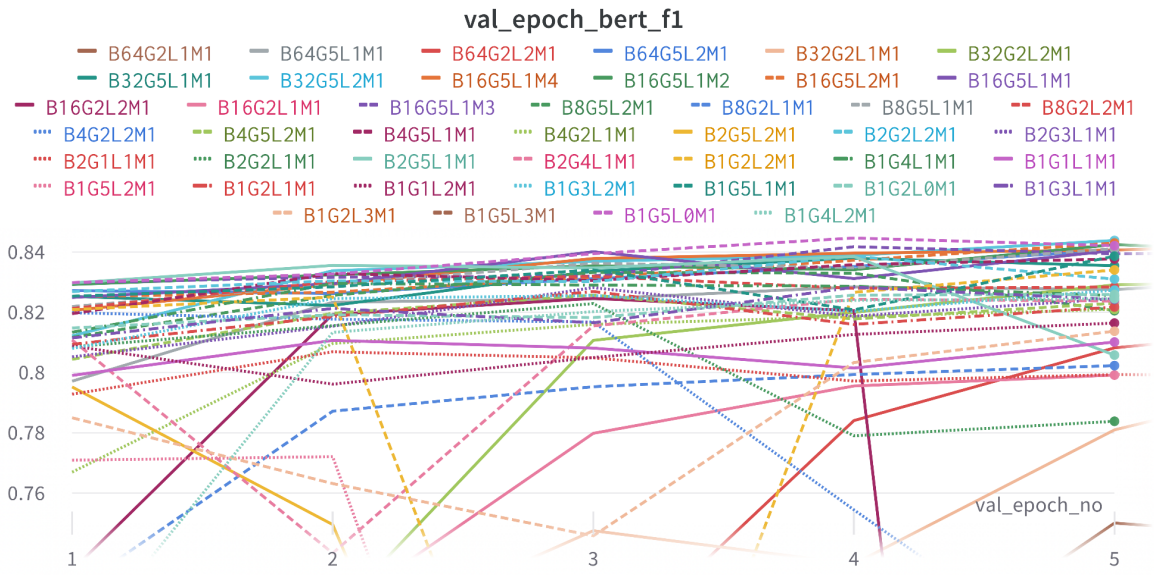

Figure15: BERT-f1scores for all the configurations of task2

In detail, the "B16G5L1M1" configuration encompasses the following parameter selections:

- Batch size of 16
- GPT-2 type: 'olm/olm gpt2 dec 2022'
- Learning rate: 0.0001
- Maximum sequence/context length: 80

Remarkably consistent BERT scores across various configurations underscore the negligible influence of maximum sequence length on the model's performance. Regardless of the sequence length, the model demonstrates strong performance. The "B16G5L1M1" configuration's consistent excellence across validation batches underscores its robustness and suitability for answering AIML queries, aligning well with the overarching objectives of our text generation task.

Table 4 showcases the BLEU, ROUGE, and BERT metric scores for the most successful configurations of the 'olm/olm gpt2 december 2022' model on the test data. The highlighted configurations exhibit robust performance during deployment, offering significant insights into the model's effectiveness based on their respective BLEU, ROUGE, and BERT scores.

Table 4: BLEU, ROUGE and BERT metric scores for task 2 best-performing configurations

Model	BLEU	ROUGE				BERT		
		R1	R2	RL	RLsum	Precision	Recall	F1
B16G5L1M1	16.04	40.15	19.20	32.28	32.18	0.84	0.84	0.84
B16G5L2M1	17.29	39.68	19.36	31.63	31.66	0.84	0.85	0.84

Conclusions

Task 1 of this project introduces an innovative methodology to tackle the intricate challenge of generating concise email subjects through the utilization of GPT-2 on the enron email dataset. Our exploration involved a systematic examination of the GPT-2 model's performance using various pretrained variants, diverse batch sizes, learning rates, and context/sequence lengths. Notably, the 'olm/olm gpt2 december 2022' pretrained variant, combined with a substantial context length, low batch size, and modest learning rate, emerges as a frontrunner in performance. This variant's superiority is attributed to its fine-tuning process, leveraging a meticulously curated snapshot from December 2022 of Common Crawl and Wikipedia. Rigorous evaluations conducted via BLEU and ROUGE metrics affirm the efficacy of our devised model. Moreover, to reinforce its capabilities, selected models underwent manual evaluation, generating subject lines for recent emails from a variety of sources. This comprehensive approach underscores the substantial potential of our methodology in effectively addressing the intricate task of generating relevant and impactful email subject lines.

Task 2, focusing on question-answering within the AIML domain, presented a significant challenge due to the absence of a specialized dataset. To successfully tackle the objectives of Task 2, we undertook the task of creating a domain-specific question-answering dataset specifically curated for the AIML domain. This endeavor was crucial in overcoming the dataset limitation and aligning with the project's objectives to effectively address Task 2 within its designated scope.

In addressing queries within the AIML domain, our project extensively examined the efficacy of the GPT2 model for question-answering tasks. We delved into the nuanced influence of various factors, including different GPT2 pretrained variants, batch sizes, learning rates, and context/sequence lengths. Through our investigations, we unearthed that the 'olm/olm gpt2 december 2022' pretrained variant, combined with a low learning rate, medium batch size, and an ample context length that accommodates the designated question and answer length, significantly impacts the model's performance. Our evaluation of the GPT2 model encompassed a spectrum of metrics, including the newly introduced BERT metric, in conjunction with the BLEU and

ROUGE metrics used previously. Additionally, to comprehensively assess the fine-tuned QA system, we conducted human evaluations alongside automated metric evaluations.

Future scope

In the realm of future prospects, expanding the scope of this project involves

- The Enron email dataset, which forms the foundation of our task 1, predominantly comprises emails originating from employees of the Enron Corporation. However, to foster a broader and more comprehensive model generalization, there is a prime opportunity to introduce a substantial influx of emails originating from diverse disciplines and domains. By augmenting the dataset with a significant volume of emails from varied contexts, we can substantially enhance the model's ability to generalize beyond the scope of Enron-specific emails.
- Enhancing the Task 2 dataset's size can significantly improve model generalization and enable multi-sentence answers, aligning with larger machine learning datasets.
- Upgrading to a more advanced computing setup than the current Google Colab 15GB GPU could potentially lead to enhanced text generation accuracy. This upgrade might also enable the utilization of more sophisticated GPT-2 variants, further elevating the quality of text generated for these tasks.

Challenges

1. The application of the latest and most effective pretrained models to the tasks at hand is limited by computational and memory constraints.
2. Training large datasets poses a significant challenge.
3. When attempting to work with larger context lengths and batch sizes, the available GPU compute becomes inadequate.
4. Model takes a substantial amount of time to carry out complete extensive hyperparameter optimization.
5. The subject lines generated fell short of reaching human-level quality standards.
6. The process of adjusting contrastive search parameters to generate an improved subject line demands a considerable amount of time.
7. Due to the model size and free-tier limitations, we could deploy the model locally.
8. Due to the scarcity of data, training results in overfitting, which in turn complicates the process of making accurate inferences.
9. During the inference the phrases in the answer are being reiterated with higher frequency after the initial sentence.
10. The answers generated by the model in response to the questions provided are not meeting expected standards.
11. In numerous instances, the model's generated responses take the form of binary "yes" or "no" answers, even when the question pertains to a different nature.

Applicability in the real world

The task email subject line generation can be considered as an abstractive text summarisation(ATS) operation, its aim is to generate a short and concise summary that captures the salient ideas of the source text. The generated summaries potentially contain new phrases and sentences that may not appear in the source text. Some possible real world applications of the task email subject line generation are:

- Marketing: Email subject line generation can help marketers craft catchy and persuasive subject lines that can increase the open rate and conversion rate of their email campaigns. It can also help them test different subject lines and optimize their performance.
- Customer service: Email subject line generation can help customer service agents write clear and informative subject lines that can address the customer's issue and expectation. It can also help them prioritize and categorize the incoming emails based on the subject lines.
- Education: Email subject line generation can help students and teachers write effective and concise subject lines that can communicate the purpose and content of their emails. It can also help them avoid spam filters and ensure their emails are read and responded to.

The domain specific question answering systems are used in multiple scenarios and across a variety of industries. Typically information retrieval use cases are best suited for question answering where there are usually one or only a few correct responses to a user question. Some possible real world applications of a domain specific question and answer system are:

- Healthcare: A domain specific question and answer system can help patients and doctors access reliable and relevant information about various medical conditions, treatments, symptoms, and medications. It can also help them find the best healthcare providers and facilities based on their needs and preferences.
- Law: A domain specific question and answer system can help lawyers and clients find accurate and up-to-date information about various legal topics, cases, statutes, and regulations. It can also help them prepare for legal proceedings and negotiations by providing relevant facts and arguments.
- Education: A domain specific question and answer system can help students and teachers learn and teach various subjects, such as math, science, history, and languages. It can also help them assess their knowledge and skills by providing feedback and explanations.

References

1. Zhang, R. and Tetreault, J., 2019. This email could save your life: Introducing the task of email subject line generation. *arXiv preprint arXiv:1906.03497*.

2. Bojic, I., Ong, Q.C., Joty, S. and Car, J., 2023. Building Extractive Question Answering System to Support Human-AI Health Coaching Model for Sleep Domain. *arXiv preprint arXiv:2305.19707*.

3. Wang, H., Li, J., Wu, H., Hovy, E. and Sun, Y., 2022. Pre-Trained Language Models and Their Applications. *Engineering*.

4. Xiang, T., Generative Question Answering for a Chatbot in the Human Resources Domain, Web article accessed with link 'https://wwwmatthes.in.tum.de/file/taw7f83f6nih/Sebis-Public-Website/Student-Theses-Guided-Research/Current-Theses-Guided-Researches/Guided-Research-Tao-Xiang/Tao%20Xiang%20GR_Report.pdf on 08.06.2023'.

5. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., 2019. Language models are unsupervised multi-task learners. *OpenAI blog*, *1*(8), p.9.

6. Sennrich, R., Haddow, B. and Birch, A., 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

7. Lin, C.Y., 2004, July. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*(pp. 74-81).

8. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y., 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.