UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Artificial Intelligence and Data Engineering

Generative AI for Social Media Post Creation

Project Documentation

Martina Marino

Roberta Matrella

Academic Year: 2022/2023

Contents

1 Introduction

In the ever-evolving landscape of social media, businesses face unique challenges in creating engaging and impactful content to connect with their target audience. Crafting captivating social media posts consistently can be time-consuming and resource-intensive. However, with the emergence of generative artificial intelligence (AI), businesses can leverage cutting-edge technology to streamline their social media post creation process and drive stronger engagement.

Generative AI refers to the use of advanced machine learning algorithms to generate original content that resembles human-created posts. By harnessing the power of generative AI, businesses can optimize their social media strategies, saving valuable time and resources while delivering high-quality content. Let's explore the transformative potential of generative AI for social media post creation specifically from a business perspective.

Generating social media posts manually can be a labor-intensive task for businesses, especially for those managing multiple accounts or posting frequently. Generative AI automates the content creation process, enabling businesses to generate posts at scale and reducing the time and effort required. This allows social media teams to focus on other critical activities such as strategy development, audience engagement, and data analysis, leading to enhanced efficiency and productivity.

Generative AI has the power to revolutionize the way businesses engage people on social media. However, it is essential to strike a balance between automation and human oversight, ensuring that the generated content aligns with the brand's values and resonates with the target audience. By embracing generative AI in social media post creation, businesses can stay ahead in the competitive social media landscape and establish meaningful connections with their customers.

1.1 Goals

The goal of this project is to leverage generative AI for the automatic generation of content for social media, particularly posts characterized by images and caption. Through the use of generative algorithms, custom images and related caption are generated from a product image for the purpose of social media advertisment.

The main objective is to provide an efficient and automated method for creating engaging and captivating content for social media platforms, thereby assisting businesses in effectively promoting their products and reaching a wider audience. By leveraging artificial intelligence, this system can generate a variety of unique and interesting images and captions that align with the brand and advertising objectives of the company.

Furthermore, the use of generative algorithms allows for great flexibility in the creative process, enabling a wide range of outcomes to cater to the diverse needs of social media advertising campaigns. This project aims to streamline the content creation process for social media, minimizing the need for human intervention and expediting the production of high-quality advertising materials.

2 Dataset

2.1 SentiCap dataset

The *SentiCap* [1] dataset contains few thousand images with captions with positive and negative sentiments. These sentimental captions are constructed by the authors by re-writing factual descriptions. In total there are more than 2000 sentimental captions. From the total of the rows only the positive ones have been considered in the following attempts because the aim is to generate a post caption with a positive meaning. The dataset is already cleaned and ready to be utilised for the new model.

Figure 1: Example of sentimental caption

2.2 Amazon dataset

The dataset used for the development of this project is the *Amazon Product Co-Purchasing Network Dataset* and is provided by the SNAP (Stanford Network Analysis Project) lab at Stanford University.

This dataset represents a product co-purchasing network on Amazon and contains information on different products, their categories and the co-purchasing relationships between them. The dataset was created using information from Amazon's website and represents a subgraph of the complete network of product co-purchase relationships.

The dataset is divided into several product categories, each of which is represented by a separate file. The files are in the format ".json.gz" and contain the data structured as follows: *asin, imUrl, description, categories, title, price, salesRank, related, brand.*

The categories taken for the following steps are `meta_Electronics.json.gz` and `meta_Office_Products.json.gz` that contain a total of 633.034 rows.

2.2.1 Preprocessing

The dataset contained fields like *asin*, *price*, *saleRank* and *related* not useful for the aim of the project, so they have been removed. The remaining fileds contained a lot of NaN values that have been removed in order to maintain only the product with all the interesting characteristics available. This led to a reduction to 180.002 rows. All the fields

except from the *imUrl* one (containing the url of the image of the product) contained a lot of HTML escape characters bad formatted, so they have been substituted with the correct format. From the resulted dataset, 10.000 rows have been used for the *text-to-text* models. In particular, for the *gpt-2 fine tuned* one some additional columns have been generated as follows:

- **title** - the concatenation of title, brand and categories of the product
- **keywords** - extracted from the description field plus the brand
- **text** - the renamed description field.

The keywords have been extracted with the use of the *spacy* library that offers some useful functionalities for the NLP processing. It loads the Spacy `en_core_web_sm` template, which supports English text processing and contains features like tokenization, POS tagging, NER, dependency analysis, and more. The spacy model is exploited to locate the relevant entities in the *title* and assign the result to the *keywords* column.

3 Models

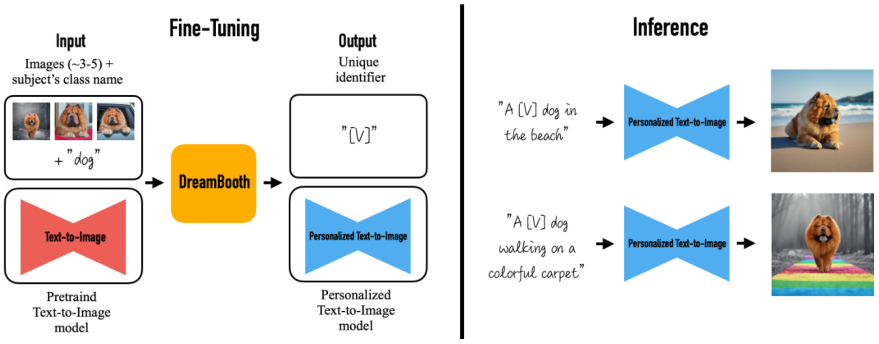
This section describes the models used as starting point for the different solutions.

3.1 Dreambooth - image-to-image

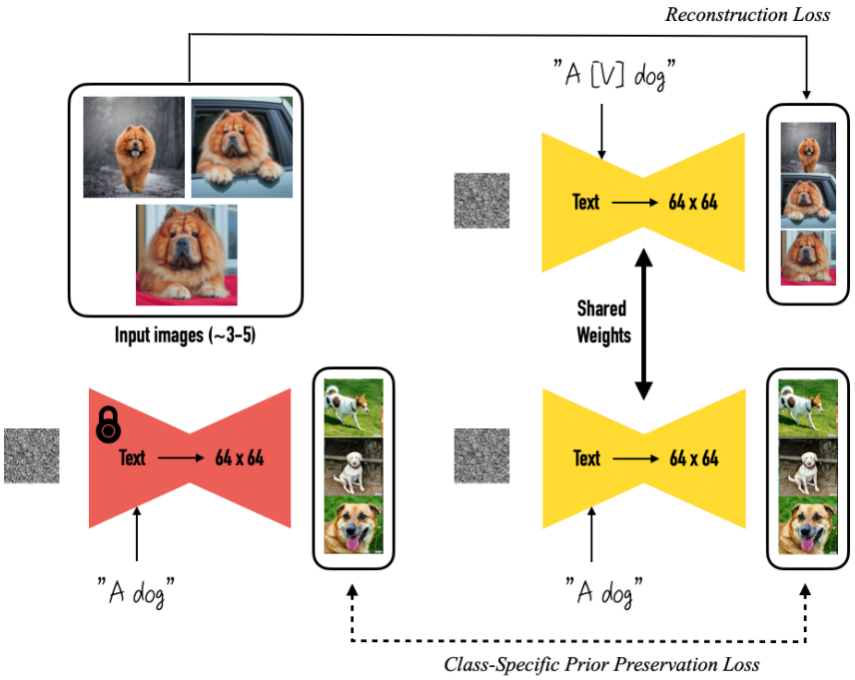
The most challenging aspect of this project is the generation of customized images. To address it, the Dreambooth model was chosen.

Dreambooth [2] is a method to customize text-to-image models like Stable Diffusion [3] given just few images of a subject. It allows the model to generate contextualized images of the subject in different scenes, poses, and views. It is a real challenge to be able to generate images that are faithful to the key elements of the subject but in different contexts with text-to-image models existing in the state of the art. Even with several iterations of a prompt that contains accurate descriptions of the object, these models are unable to reconstruct its key features. In fact, the other models, while able to generate changes in the subject, are unable to reproduce it accurately and in different contexts.

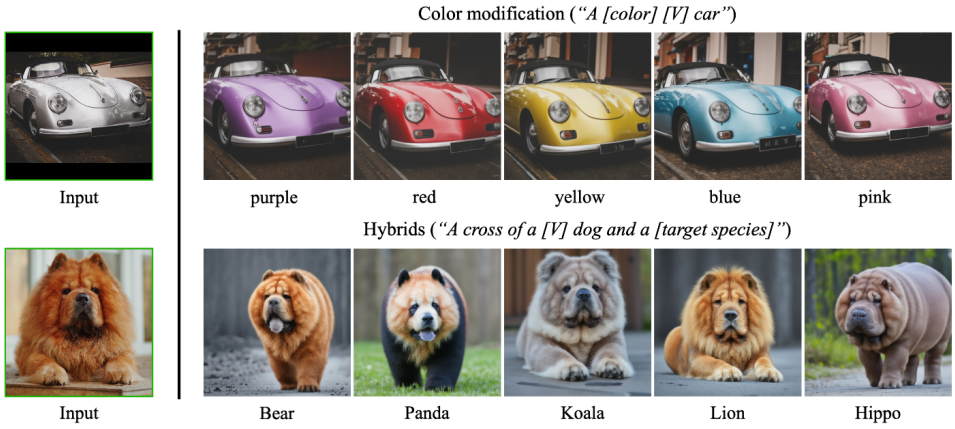
Dreambooth takes as input few images of a subject and the corresponding class name and returns a fine-tuned text-to-image model that encodes a unique identifier that refers to a subject. At inference time this identifier can be used in different sentences to obtain the subject in different contexts.



During the fine-tuning, in parallel with the images and text prompt containing the unique identifier and the name of the class of the subject, a class-specific prior preservation loss is applied. It gives higher semantic priority that the model has on the class, promoting the generation of different instances of the subject's class injecting the class name in the text prompt.



This model can perform property modification such as color modifications and hybrids but still preserving the unique visual features that give the subject its essence.



Dreambooth allows also to give original artistic renditions with some creativity and to put some accessories to the subject in a realistic and various way.

3.2 Vision Transformer (ViT) - image-to-text

Vision Transformers have extensive applications in popular image recognition tasks such as object detection, segmentation, image classification, and action recognition. Moreover, ViTs are applied in generative modeling and multi-model tasks, including visual grounding, visual-question answering, and visual reasoning.
Vision Transformers (ViT) is an architecture that uses self-attention mechanisms for

image processing. Each transformer block consists of two sub-layers: a multi-head self-attention layer and a feed-forward layer.

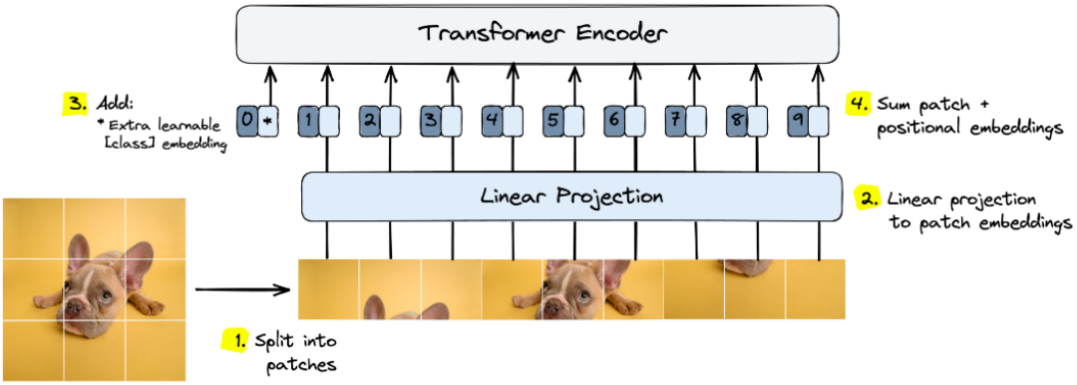


Figure 2: ViT architecture

The attention weights for each pixel in the image are calculated by the self-attention layer based on the relation of the pixel with the other, while the feed-forward layer applies a non-linear transformation to the output of the self-attention layer. Multi-head attention extends this mechanism by allowing the model to deal with different parts of the input sequence at the same time.
The main steps performed by ViT are the following:

- **Patch Extraction**: The input image is divided into fixed-size non-overlapping patches. Each patch represents a local region of the image and contains a fixed number of pixels. These patches are then linearly flattened into sequences.
- **Position Embedding**: Each patch is associated with a learnable position embedding vector. Position embeddings provide positional information to the model, enabling it to capture spatial relationships between patches.
- **Tokenization**: Similar to natural language processing, each patch is treated as a token, and an additional learnable token embedding is added to represent the image-level information. These tokens, along with their position embeddings, form the input sequence for the Transformer model.
- **Transformer Encoder**: The Transformer encoder consists of multiple layers, typically comprised of a multi-head self-attention mechanism and feed-forward neural networks. The self-attention mechanism allows each patch/token to attend to all other patches/tokens, capturing global dependencies and enabling the model to learn contextual representations.
- **Classification Head**: The output of the Transformer encoder is passed through a classification head, typically a linear layer, to produce the final predictions. In the case of image classification, the classification head maps the encoded sequence into class probabilities.

3.3 GPT-2 - text-to-text

GPT-2 [4] is a transformer model pretrained on a vast corpus of English text using a self-supervised approach. This means it was trained on raw text data without any human labeling, allowing it to leverage publicly available data. The training process involved generating inputs and labels from the text, specifically by predicting the next word in sentences. To be more specific, the model takes in sequences of continuous text with a certain length as inputs, and the targets are the same sequences but shifted one token (word or subword) to the right. The model internally employs a masking mechanism to ensure that predictions for a given token only using the inputs from positions earlier in the sequence, excluding future tokens. By training in this way, the model learns an internal representation of the English language, which can then be utilized to extract features that are valuable for downstream tasks.

The simplest way to run a trained GPT-2 is to allow it to ramble on its own (which is technically called generating unconditional samples) – alternatively, it is possible to give a prompt to have it speak about a certain topic (a.k.a generating interactive conditional samples). The model only has one input token, so that path would be the only active one. The token is processed successively through all the layers, then a vector is produced along that path. That vector can be scored against the model's vocabulary (all the words the model knows, 50,000 words in the case of GPT-2). In this case, we selected the token with the highest probability. GPT-2 has a parameter called top-k that we can use to have the model consider sampling words other than the top word. Subsequently, the output of the first step is added to the input sequence and the model makes its next prediction. More in detail, GPT-2 is characterized by:

- **Transformer Encoder**: GPT-2 consists of a stack of multiple identical layers, known as the Transformer encoder. Each encoder layer consists of two sub-layers:
  - Masked Self-Attention Mechanism: This mechanism allows the model to weigh the importance of different words in a sentence based on their context. It captures dependencies between words and enables the model to understand long-range relationships. It does that by assigning scores to how relevant each word in the segment is, and adding up their vector representation
  - Feed-Forward Neural Network: After the self-attention mechanism, a simple position-wise fully connected feed-forward neural network is applied to each word's representation independently.
- **Layer Normalization**: Both the self-attention mechanism and the feed-forward network have residual connections around them. Layer normalization is applied after each sub-layer to normalize the outputs and stabilize training.
- **Positional Encoding**: To capture the sequential order of words in a sentence, GPT-2 utilizes positional encodings. These encodings are added to the word em-

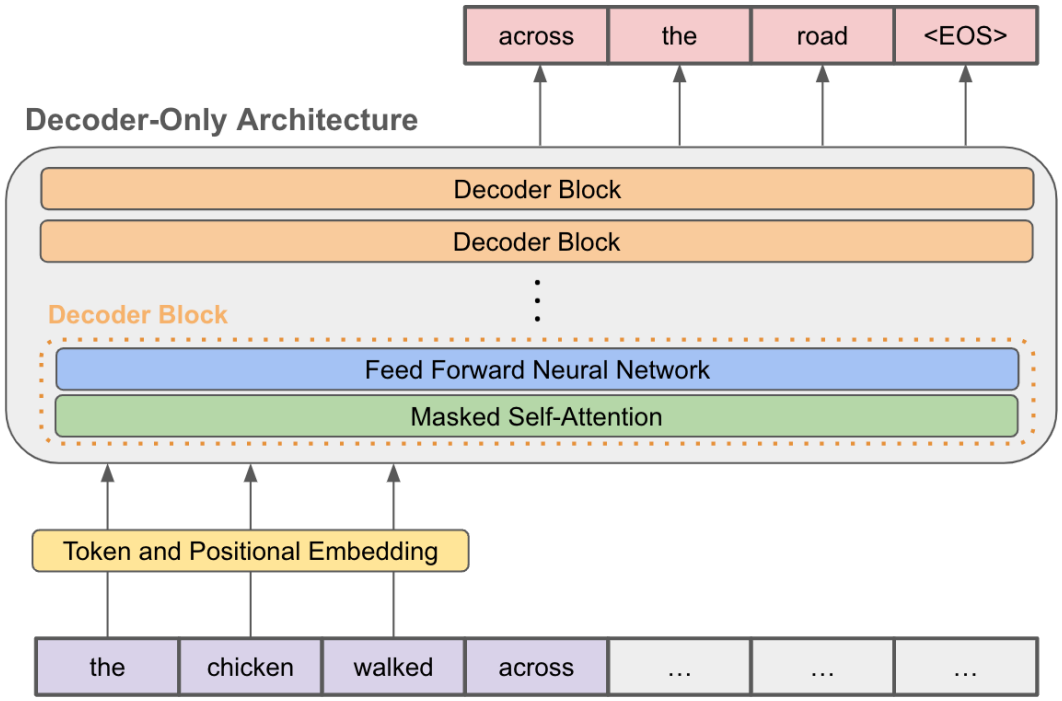


Figure 3: GPT-2 architecture

beddings and provide the model with information about the relative positions of the words in the input sequence.

- **Token Embeddings**: Each word in the input sequence is represented as a vector, known as a word embedding. GPT-2 uses a large fixed-size vocabulary and maps each word to a dense vector representation. The model looks up the embedding of the input word in its embedding matrix.

Figure 4: Use of Token Embeddings and Positional Encodings

The GPT-2 architecture is designed to generate coherent and contextually appropriate text by leveraging the self-attention mechanism and the contextual information captured through pre-training. It has achieved impressive performance on a wide range of natural

language processing tasks and has been widely adopted for various text generation and understanding applications.

3.4 T5-base - text-to-text

The developers of the Text-To-Text Transfer Transformer (T5) write: "With T5, we propose reframing all NLP tasks into a unified text-to-text-format where the input and output are always text strings, in contrast to BERT-style models that can only output either a class label or a span of the input. Our text-to-text framework allows us to use the same model, loss function, and hyperparameters on any NLP task." T5-Base, the version used for this purpose, is the checkpoint with 220 million parameters.

This is a sequence-to-sequence model that can do language translations, text summarization or "translations" of texts written in a certain style to another (e.g., formal to casual or Shakespearean English to modern English). Given a corpus of two versions of a text, the model will learn how to return the new text with the style wanted as done by the causal language modeling.

The T5 model has achieved state-of-the-art performance on various benchmark NLP tasks and has been widely adopted and used in research and industry applications. In this project this model was used to transform a neutral and descriptive text into a positive text motion more suitable for advertising purpose.

4 Fine tuned models

The descripted models are the starting point to made up a model that fits the selected dataset but it also able to generalize concepts. To obtain such a network, the fine tuning is necessary to better teach the model to recognise the specific items present in the specific dataset.

4.1 ViT-GPT2 with Amazon dataset

The first experiment has been performed using an image-to-text model in order to extract captions directly from the images. A preliminary state-of-the-art study have been performed for determine both the model most suitable for the captions generation, both the metrics to evaluate it. The most suitable model for caption generation is the VisionEncoderDecoder. The VisionEncoderDecoderModel is an image-to-text model that combines the characteristics learned by a Transformer-based vision model (encoder) with the language comprehension skills of a pre-trained language model (decoder).The VisionEncoderDecoderModel can be used to initialize an image-to-text model with any pretrained Transformer-based vision model as the encoder (e.g. ViT, BEiT, DeiT, Swin) and any pretrained language model as the decoder (e.g. RoBERTa, GPT2, BERT, DistilBERT). This model have been chosen because it is able to automatically identify the objects in the image and generate the corresponding caption. In order to achieve this result, have been chosen ViT as encoder and gpt-2 as decoder.

The Amazon dataset previously described has very long descriptions and for this reason is not suitable for the fine tuning of ViT-GPT2 model as it is. In fact, it raises a CUDA out of memory error because of the unsufficient resources at disposition. Furthermore a truncation of the sentences is not an option because it would led to a meaning loss and also a summarization technique it is not appropriate due to the nature of the description that is made of technical characteristics of the product. So, other solutions have been explored as follows.

4.2 ViT-GPT2 with Amazon dataset positive version

A first attempt done is the fine tuning of the ViT-GPT2 model using a dataset made of the images of the Amazon dataset product and the relative "positive" caption. These captions have been generated using a fine tuned Seq2Seq model using *T5-base*. For this purpose, the SentiCap dataset have been used. Considering that the captions contained in this dataset are related to a subset of the validation set of the *COCO dataset* version 2014, the idea was to generate a new labeled dataset for the *Sequence-to-Sequence* model with a correspondence between the neutral caption generated by the *ViTGPT2* model and the positive version of it. For this purpose, the `COCOval2014.zip` file has been downloaded from the source site and the `dataset.csv` file has been generated. Note that only the captions with the positive sentiment have been considered for this aim and that

there are about 3 different positive-version sentence for each neutral one. Later, T5-base has been fine tuned on the neutral-positive pairs of sentences in order to obtain a model able to take in input a neutral and descriptive caption and give it a more positive meaning adding new adjectives and expressions.
An example of execution of the fine tuned model is showed below:

```
inputs = ["Messenger bag with zippered accessory pockets to store cables,
power bank, pens and other accessories; ideal present for men"]

output:
- that is a great item to present to your loved one, family friend,
  friend or partner.
- this nice bag has a wide, padded interior and a zipper divider in
  the back
- a stylish messenger bag with lots of storage pockets for important
  items
- this is a great size large messenger bag for traveling
- this bag would be an excellent gift to the guys
```

This model have been exploited in order to generate the "positive" descriptions for the Amazon dataset. The so generated descriptions have been given in input with the correspondent images to the *ViTGPT2* model.

The result is not very good because that model is not able to generalize since the amount of data necessary for a satisfactory fine tuning is very high and requires the downloading of the images in local in order to perform it. The captions produced by this model have a positive connotation as desired but they are still too generic and do not report the technical characteristics of the products. Here are some examples of execution:

Figure 5: Images in input

```
Output 1 : "a great way to get great images to your camera is to use
            filters that allow for better light to reach"
Output 2: "a nice card for storing your favorite music on"
```

4.3 GPT-2 with Amazon dataset

The main problem to address was the generation of a description coherent with the context. GPT-2 is trained using a language modeling objective, which means it is optimized

to predict the next word in a sentence given the context. This makes GPT-2 particularly effective for generating coherent and contextually relevant descriptions.
In this section are described the main steps of our customized fine tuning of the pre-trained Hugging Face GPT-2 [5]. The previously described preprocessed Amazon dataset was splitted with a percentage of 80 for the training set and 20 for the validation set. In standard text generation fine-tuning, since the next token is predicted given the text, the labels are only the shifted encoded tokenized input. In order to generate caption that were related to the product to be advertised, training of the model have been carried out by exploiting not only the descriptions but also the title and the keywords extracted from them. As input to the model, therefore, not only the description have been given, as is generally the case, but also the other additional fields previously reported, separated by special tokens. In addition, to ensure good generalization of the model, data augmentation was used by sampling and shuffling the list of keywords during training. No augmentation has been applied during validation to retain consistency across epochs. In addition, in order to perform the best possible fine tuning by taking advantage of the limited resources provided by Google Colab, it was decided to unfrozen the last 6 layers and use a batch size of 2 and an epoch number of 2.

For the caption generation two different decoding methods have been compared.

1. **Beam search** Beam search reduces the risk of missing hidden high probability word sequences by keeping the most likely num_beams of hypotheses at each time step and eventually choosing the hypothesis that has the overall highest probability.

2. **Top-p sampling** Top-p sampling chooses from the smallest possible set of words whose cumulative probability exceeds the probability p. Then the probability mass is redistributed among this set of words. In this way, the size of the set of words can increase and decrease dynamically according to the next word's probability distribution.

In both cases the max length of the text have been set to 75 in order to get a small size text, more suitable for a social media post. In beam search the top_k parameter, that fixes the size of the list of the high-scoring tokens, was put at 30 and the parameter top_p, that is the shortlist of the top tokens whose sum of likelihood exceeds a certain value, was set to 0.7. The temperature at 0.9 represents a low randomness of the text generation. In the beam search, the parameter used for group beam search, num_beams, was placed at 5 in order to archive a trade-off between precision and speed. An high repetition penalty have been set in order to reduce the probability to generate multiple times the same words.

4.4 Dreambooth with Amazon dataset

The model was trained by providing some images, approximately between 3 and 5 from different angles and perspectives, of the desired products and a prompt. The prompt,

defined for each product, is characterized by an instance prompt, within which there is a specific identifier for the product and a generic identifier for the class of the product and a class prompt where the generic class of the product is specified. The prompts are saved into the `concepts_list.json` file. The images to be used for fine tuning can be provided by drag and drop or by manually inserting them in the appropriate folder.

```
[
    {
        "instance_prompt": "photo of a bg pc bag",
        "class_prompt": "photo of a pc bag",
        "instance_data_dir": "/content/drive/MyDrive/BPM/Dreambooth/img/bg",
        "class_data_dir": "/content/drive/MyDrive/BPM/Dreambooth/output/bg"
    }
]
```

Figure 6: Prompt example for a PC bag

The choice of parameters was made considering the resources provided by Google Colab. Since the resources required by this model are very high and those provided by Colab are limited, the configuration of parameters shown in the first line of the figure 7 has been chosen.

fp16	train_batch_size	gradient_accumulation_steps	gradient_checkpointing	use_8bit_adam	GB VRAM usage	Speed (it/s)
fp16	1	1	TRUE	TRUE	9.92	0.93
no	1	1	TRUE	TRUE	10.08	0.42
fp16	2	1	TRUE	TRUE	10.4	0.66
fp16	1	1	FALSE	TRUE	11.17	1.14
no	1	1	FALSE	TRUE	11.17	0.49
fp16	1	2	TRUE	TRUE	11.56	1
fp16	2	1	FALSE	TRUE	13.67	0.82
fp16	1	2	FALSE	TRUE	13.7	0.83
fp16	1	1	TRUE	FALSE	15.79	0.77

Figure 7: Table to choose parameters for Dreambooth training

The basic parameters set were *steps* and *guidance scale* which respectively indicate the quality level of the image and how closely the AI should follow your prompt. Generally *steps* over 25 generates high quality level images but higher steps makes the process slower. Moreover, the *guidance scale* can reach maximum value of 20 but with worse image quality; indeed a value between 7 and 12 is used to achieve the best trade off between artistic results and follows of the prompt [6]. Starting from these values, after several tests, we obtained a compromise between creativity and adherence to the prompt by choosing values 70 and 8. Moreover, the images have been generated of 512x512 pixel using a specific prompt. In the prompt is specified the unique identifier of the product, its class name and the characteristics wanted, as a specific color or a specific location. A negative prompt have been used too in order to avoid the generation of distorted or out of frame images. Indeed, the negative prompt specifies what must not be present in the figure. The words specified in this prompt have been chosen between the most used for

the negative prompt:

negative prompt = ugly, tiling, mutated hands, fused fingers,extra fingers, poorly drawn hands, poorly drawn feet, poorly drawn face, out of frame, extra limbs, disfigured, deformed, body out of frame, bad anatomy, watermark, signature, cut off, low contrast, underexposed, overexposed, bad art, beginner, amateur, distorted face

Indeed, one of the limit of Stable Diffusion is that it frequently generates images with anomalous number or shape of arts and fingers. The negative prompt helps reducing this problem. A similar problem regards the text because Stable Diffusion haven't been trained on images of text.

5 Examples of post generation

Given the models described in the previous sections, the phase of the post creation follows. The starting point is the product to be advertised with its characteristics and the idea to be communicate with it.

Some of the different examples of test performed are reported below.

Minolta camera

Given the following images

Figure 8: Starting images for Dreambooth fine tuning

and the following prompt with *cm* as unique identifier and *camera* as class identifier

```
prompt : photo of a cm camera on the sand
```

the following image have been generated:

Figure 9: Image generated with Dreambooth for the camera of minolta brand

As it is possible to observe, the model returns an image similar to the ones previously provided but in the requested background. The image is realistic,the shadow corresponds to the camera and the light.

The captions have been generated with gpt-2 using the title

```
enjoy your holidays with a camera
```

and the keywords

```
minolta, camera, photos
```

The caption have been generated exploiting the previous described models:

- gpt-2 with top-p text generation:

```
 The perfect vacation is one that takes the best moments of any time.  This stylishly designed digital zoom lens combines high-performance optics and an excellent viewfinder to create vibrant images in all conditions: sunlight or shade on sunny days

```

- gpt-2 with beam-search text generation:

```
 The world of digital photography is filled with great pictures and great sound. You can take the time to enjoy those moments with this compact Hi-Fi camera. It features a 2.5mm f/2.8-3.5 zoom lens

```

both results give a representation of a possible caption for the image even if the one generated with top-p text generation is more in line with the context.

Tripod

As in the previous experiment some images have been provided to the Dreambooth model for the fine tuning

Figure 10: Image generated with Dreambooth for the tripod

and the following prompt with *tr* as unique identifier and *tripod* as class identifier

```
prompt: photo of a tr tripod in a starry night
```

the model have generated images like the following:

Figure 11: Image generated with Dreambooth for the tripod

Also in this case the images generated by the model are satisfying because the product has been reproduced well and also the background is coherent with the requested one. The caption for this product have been generated using gpt-2 using the two different decoding methods with the title

```
use your tripod for the outdoor trips
```

and the keywords

```
tripod, outdoor
```

- gpt-2 with top-p text generation:

```
The most important thing you can do to make sure that  a great
   vacation is not lost in its details. It's also an easy way of
   putting it on topofthe world with this design and one-time
   operation,,a perfect fit if traveling through any environment or
    even just

```

- gpt-2 with beam-search text generation:

```
The Outdoor Tripod is an ultracompact, lightweight, easy-to-use
   tripod. It features a 2.5-inch f/2.8-3.5-5.6  zoom lens and a
   1.25-megapixel digital camera

```

The first method does not return a caption too attractive while the second is already more suitable.

Computer bag

A last example is the one shown below. Again, some images for fine tuning were given to Dreambooth:

Figure 12: Image generated with Dreambooth for the pc bag

with the prompt characterized by *bg* as unique identifier and *pc bag* as class identifier

```
prompt: photo of a bg pc bag on an office chair
```

the model have generated images like the following:

Figure 13: Image generated with Dreambooth for the pc bag on a chair

The image obtained is adequate compared to the prompt used even if it does not specific in some details that, exploiting more resources and improving some parameters, could be made in a more realistic way. The captions have been generated with the two different encoding approaches using as title

```
a suitable pc bag for office
```

and as keywords

```
pc bag, leather
```

- gpt-2 with top-p text generation:

```
    The Case Logic C-550 Portable PC Bag is a lightweight and
stylish  wallet. The case features an ergonomic design that
holds up to the most serious needs of your digital

```

- gpt-2 with beam-search text generation:

```
 The PDA-P is the world's smallest and most affordable PC bag.
The PDA-P is made of durable plastic and has a high quality
leather interior

```

The caption regards two different models of leather pc bags that are present in the Amazon dataset.

Despite having a great limitation of resources to do the training, with very modest dataset and the most limited version of the model, quite acceptable results have been achieved, which demonstrates the potential of GPT-2. The maximum token length of the captions generated have been set to 75 but it can be changed based on the social and the type of post.

6 Conclusion

In conclusion, the work described in this documentation has provided satisfying results for the generation of social media posts for product advertising. It is clear that the results obtained do not represent the final and complete version of the post and, therefore, that a phase of analysis and correction of the caption and image would still be necessary. As already mentioned in the introduction, the aim is to provide help in designing an advertising post and, above all, to realize a preview of how the actual post could look like. Furthermore, the results obtained could be used as inspiration to create a photo shoot that reproduces the same results or simply use a photo editing software to obtain the final post.

For what concern images, in fact, they are not precisely reproducing the product given in the input photos and the major problem regards the logos and the writings on the product that the model cannot reproduce correctly.

Moreover, due to the restricted resources offered by the free *Google Colab* software, the text-to-image model exploited in this project has the minimal computational and storage requirements found in literature. For this reason, the results obtained are satisfying but surely with a more powerful model and more training time more accurate images could be generate.

References

[1] Alexander Patrick Mathews, Lexing Xie, and Xuming He. Senticap: Generating image descriptions with sentiments. *CoRR*, abs/1510.01431, 2015.

[2] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

[3] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[4] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[5] Ivan Lai. Conditional text generation by fine tuning gpt-2. 2021.

[6] *getimg.ai*.