POLITECNICO DI TORINO

Bachelor's Degree in Computer Engineering

Internship Report

# Automation of Product Description Generation

Company Tutor

**Michele SONNESSA**

Academic Tutor

**Luca CAGLIERO**

Student

**Niyousha NAJMAEI**

**s270244**

**Oct 2021**

# Summary

In the highly-competitive market of e-commerce, having good product descriptions is essential for rising above the competition as good product descriptions increase the probability of purchase substantially. This is because, e-commerce platforms highly depend on textual recommendations and product descriptions for convincing the customer to purchase items, a task that in physical stores is achieved by the persuasion techniques of the salespeople. Writing these descriptions, however, is a dull and seriously time-consuming task. Therefore, automating this process has been of interest in both academic literature and industry.

During my internship at Zero 11, I tackled the feasibility of using Pre-trained Language Models for generating appealing, informative and accurate product descriptions, while finding a way to minimize the use of resources in order to maximize profit to expense ratio when the model is used in production without significantly sacrificing the quality of the generated descriptions.

The results of the experiments were promising, and cast light on a myriad of possibilities for future work.

# Table of Contents

# List of Tables

# Acronyms

**API**

Application Programming Interface

**B2B**

Business to Business

**NLP**

Natural Language Processing

**NLU**

Natural Language Understanding

**PTM**

Pre-trained Language Model

**RL**

Reinforcement Learning

**RNN**

Recurrent Neural Network

**SEO**

Search Engine Optimization

**SKU**

Stock Keeping Unit

**TF**

TensorFlow

| | |
|---|---|
| **TPU** | Tensor Processing Unit |
| **VM** | Virtual Machine |

# Chapter 1

# Introduction

The goal of my curricular internship at Zero 11 S.r.l. was to investigate Pre-trained Language Models and the feasibility of using them for automating the process of product description generation. The internship was carried out from Sep 2021 to Oct 2021, for a total of 250 hours.

## 1.1 About Zero11

Being a Software-as-a-Service company, Zero 11 aims to facilitate selling and earning through digital distribution channels for all Italian B2B e-commerce companies, thus making it possible for them to compete on fierce national and international markets effectively. Zero11 creates business models to sell goods and services on international markets with a focus on simplicity and innovation. This is made possible thanks to their thorough knowledge of the IT field as well as the corporate business processes. This combined with their aspire to keep up with and implement latest technologies has been the key to their success.

## 1.2 About the Project

Over the years, NLP research has been moving towards the use of deep neural networks as neural language models such as BERT [1] and GPT-3 [2] rather than being heavily influenced by theories of linguistics as it was in the past. As stated in [3], defining language models as probability distributions over sentences is more effective than thinking about them as definitive sets defined by grammars, an approach used for formal languages.

Recent works have shown that Pre-trained Language Models, which have been trained on a large corpus of data, can learn universal language representations, which are beneficial for downstream NLP tasks and can avoid training a new model from scratch [4]. These large models with millions and billions of parameters are trained on large corpora of unlabeled-data to avoid over fitting. Training on unlabeled data has the advantage that there's not much cost in preparing the data set, and could be useful for the model to learn the linguistic rules, lexical meanings, syntactic structures and semantic roles. OpenAI has proven large scale training to be effective in their research during the past few years [2, 5, 6], and this idea was taken even further by Google Brain's Switch Transformer [7].

PTMs can be used for many downstream NLP tasks like Question Answering, Sentiment Analysis, Machine Translation, Summarization and Text Generation. The goal of our project was to leverage the capabilities of PTMs for automating the process of generating product descriptions. Undoubtedly, appealing, informative and accurate product descriptions are essential in today's highly-competitive market of e-commerce. Good product descriptions, which substitute effective assistance of salespeople in physical stores, could increase sales substantially. Writing these descriptions, however, is a dull and seriously time-consuming task. Thus, EleutherAI's two open source pre-trained language models, GPT-J-6B and GPT-Neo-1.3B were fine-tuned and used in this project to investigate automating the process of generating product descriptions.

## 1.3 About the Results

The experiments lead to more than 80% of the generated descriptions being sufficiently fluent, almost accurate and appealing to the reader. However, the process still needs human supervision to check the results at the end, as there's no guarantee that the model won't generate inaccurate, incomprehensible, illogical or even offensive results. Moreover, the model's performance was evaluated by generating descriptions for one of company's clients' new set of products. The result of the evaluation was promising as the client was satisfied with the generated descriptions.

# Chapter 2

# The Data Set

## 2.1 Input Files

Throughout the experiment, the input files were normal catalog exports from two of the company's clients, which in this report will be called Client-1 and Client-2. The files contain the product tags indicating product features, and the product descriptions already available in the catalogs which in almost all cases was a list of the features of the product. An example of the list-like product descriptions from the catalogs is as follows:

Wallet, eco-leather, logo
Credit card holder, document holder and coin purses
Zip fastening
Dimensions: 19,5*10,5*3 cm

## 2.2 Pre-processing

Since in most cases product tags were poor, non-exhaustive and partially describing product features, related tags, like material, had to be extracted from the list-like product descriptions to enhance the quality of product tags in the examples fed to the model. This was done for experimental purposes and in the future when descriptions have to be generated for new, never-before-seen products, which might not have a list-like product description, the clients have to be asked to provide exhaustive product tags.

More detailed tags caused the model to produce significantly better, more accurate, better structured and less outrageous descriptions. The most general set of tags present in almost all instances were brand, country of origin, category, season, color and gender.

However, in most cases, tags like sleeves length, pattern, type of neckline, information about the pockets and product material were also fed to the model.

Moreover, as the models were pre-trained on a data set with more than 95% of the data being in English, it wasn't recommended to use the models with any language other than English. Thus, all the tags had to be translated to English. The necessity of English tags was also checked during the experiments, and having only a few Italian product details, had a severe negative impact on the results.

## 2.3 Format

The format in which the prompts were written was also important. For instance, overuse of "[]{}" characters, caused the model to confuse the prompt with a JSON object and would generate results related to JSON objects rather than a product description.

Moreover, the presence of trailing spaces confused the model and all the files had to be written with no trailing new line characters, i.e. using `print(txt,file=f,end=´´)`

## 2.4 Output Files

Outputs are presented in .xlsx files containing product URLs, the tags used for the generation, generated product descriptions and the list-like description from the catalog, if available.

# Chapter 3

# Evaluation Guideline

## 3.1 The Guideline

For evaluating the generated descriptions manually a similar approach to that of [8] was taken. The descriptions were graded in two different areas, "Grammar and Fluency" and "Coherence and Consistency". Each description was given a point from 1 to 5, a description graded one being the least qualified.

"Grammar and Fluency" tests whether the result is acceptable as a natural language text, and the grammar is correct, whereas "Coherence and Consistency" checks if the description is relevant to the product, there are no inconsistent information given and all the key features of the product are communicated.

The descriptions are graded in each area by the following guideline.

- Grammar and Fluency:
  - 5: Perfectly fluent and without any grammatical errors
  - 4: Mostly Fluent and/or has at most two minor grammatical errors that do not affect understanding, e.g. *This Pants is the perfect choice for a formal occasion.*
  - 3: Semi-fluent and has serious grammatical errors that do not have a strong impact on understanding, e.g. *The dress blue has a round neck*
  - 2: Doesn't convey a clear message but it is still in the form of human language, e.g. *The print is done in a gradient, with the flowers in red on the right and the flowers in blue on the left.*

  - 1: Non-sense composition of words and/or not in the form of human language, e.g. ***A relevance of the brand to the fashion industry, the product to the daily life.***

- Coherence and Consistency:
  - 5: Accurate description with perfectly correct details about the product
  - 4: One minor wrong detail about the product (e.g. Cotton blend instead of Cotton, suitable for spring instead of spring and summer) or does not cover some minor details
  - 3: Covers part of the details and/or has minor mistakes or one severe mistake e.g. wrong color or polyester instead of cotton
  - 2: Correct category but fails to cover most of the details or contains more than one major mistake, e.g. correct category of product but incorrect gender and material
  - 1: Wrong category of product or not a product description

Note that:

- Information that wasn't present in the tags used for inference for which an assumption was made by the model, was considered equal to wrong information during the evaluations, i.e. assumptions, even if they may be correct, were considered incorrect. To reduce the amount of time consumed to rate the descriptions.
- Grammar and coherence were considered as individually as possible, i.e. if the product features were mentioned in the product but there were severe grammatical error making the understanding of the description difficult, the "Coherence and Consistency" points were given in full whereas points were subtracted from "Grammar and Fluency".
- Results in the form of a comment on a product made by a customer were considered not a product description and given 1 points for coherence despite technically describing the product. e.g. ***I absolutely love this pair of jeans, they are very comfortable and light, the pockets are very practical and the material is very soft. The design is very classic and elegant.***
- If cutting off the description at a point would make it more fluent this was considered as done before evaluating, i.e. if the model fell

**into a loop and reproduced a portion of the description over and over, the result before the repetitions were considered, similarly if from a point on in the description, non-sense was produced, this part wasn't considered in the evaluations.**

- **Descriptions using excessively repetitive words and identical sentences were given a lower fluency points.**

# Chapter 4

# Using GPT-J

As a first step, the goal was to investigate using GPT-J-6B, an open source JAX-based transformer language model developed by EleutherAI with 6 billion parameters to automate the generation of product descriptions for an e-commerce website, in order to get a better understanding of the required resources and the acceptability of the results.

## 4.1 About GPT-J

GPT-J is an open source GPT3-like PTM developed by EleutherAI in an attempt to compete with OpenAI and their funder, Microsoft. The model is not only comparable to the GPT-3 model of a similar size (6.7B GPT-3 or Curie), but being trained on a data set that contains data from GitHub and StackExchange, it also tends to outperform Curie in some tasks like code generation.

The weights of GPT-J have a size of about 61GB, which obviously won't fit on Google Colab's memory, so Google Cloud TPUs have to be used for fine-tuning this model. However, a "slim" version of the weights is also available which is a distilled version using model pruning, not containing all the parameters. Not only the inference time using the "slim" version of the weights is significantly less than that of the full-weights, but it can also fit on Google Colab's memory.

As stated in [9] when pruning BERT, removing 30-40% of the weights doesn't affect the inference quality on downstream tasks, whether weights are pruned before or after fine-tuning. Similarly for our case, the slim weights' performance for inference is not noticeably worse compared to the full ones. Fine-tuning the slim weights however, requires more warm-up steps before training with the regular learning rate to avoid forgetting the pre-trained information.

## 4.2 Fine-tuning GPT-J

Fine-tuning had to be done using a TPU VM on Google Cloud and the mesh-transformer-jax API for training the model, in order to understand the amount of required resources. The data set used for fine-tuning had to be in .tfrecords format so a python script was written to transform .xslx catalog export files containing labeled examples to .tfrecords. 25% of the data was used as the validation set and 75% as the training set, and after uploading the weights and the data set to a Google Cloud Bucket, the model was trained.

Given a semi-large data set with 1-2K instances of data, the training took too long and considering the high price of Google Cloud TPUs and the need to repeat this process in order to experiment with results and improve them, this solution was found to be too expensive to further delve into.

## 4.3 Inference with GPT-J

As inference with the fine-tuned model using Google Cloud TPUs was deemed too expensive, GPT-J without fine tuning was tested for generating product descriptions. TOP-P and temperature, two modifiable parameters controlling the randomness of the results were set to 0.5 and 0.8 respectively.

A total of 3146 product descriptions were generated for catalogs from Client-1 and Client-2 websites, of which 1345 were acceptable, 43% of the total generated descriptions. Acceptable results were the ones that had more than 2.5 points in average, and were in the form of a product description, i.e. results scoring perfect in "Grammar and Fluency" while not being a product description, thus getting a 1 in "Coherence and Consistency" weren't considered acceptable, despite scoring 3 in average. Furthermore, the average points of 100 randomly chosen instances of the generated descriptions in "Grammar and Fluency" was 3.96/5, whereas that of "Coherence and Consistency" was 2.49/5.

**Although the acceptable results constituted less than a third of all generated descriptions, the quality of these descriptions were quite impressive. A few randomly chosen instances of generated descriptions and a few of the acceptable ones are listed in Table 4.1 and Table 4.2 respectively.**

**The results generated by GPT-J without fine-tuning were used as a base-line for future results.**

## 4.4 Alternative Approaches

**A way forward could have been to investigate other hardware options, or alternatively use a lighter model that could fit on Google Colab TPUs.**

**The slim-weights of GPT-J which took around 5-20 seconds to run on Colab TPUs, using about 12GB of RAM, needed 62.1 GB of RAM and took 2 hours to generate a single product description on 1 core of a virtual server CPU, whereas the same process on a GPU needed more than 60 GB of RAM and 24GB of GPU VRAM which wasn't available at the company to test the model on. Furthermore, running the model in half precision on a GPU with 12 GB VRAM is possible as well.**

**As a way forward, GPT-Neo was used to generate descriptions.**

| **Generated Descriptions** |
|---|
| *this wallet is a very cute and practical wallet for you, it is a good choice for you to carry your money, wallet, card, mobile phone and other things.* |
| *The Carrera Sunglasses are the perfect way to add a touch of sophistication to your style. They are made of acetate and have a UV coating to protect your eyes from the sun. This pair of sunglasses is made in Italy.* |
| *luxury designer shoe with wedge design. This is a very nice and comfortable shoe. The leather is very soft and it feels great. The quality is good and the price is very good.* |
| *Furla - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet - Furla Wallet -* |
| *A beautiful leather bag with two compartments, one with a zipper and the other with a magnetic closure. In a dark brown color, this bag is very versatile. It is perfect for a daily use, for example, for shopping or to carry the daily necessities. This bag is perfect for both men and women.* |
| *Emilio Pucci madein: Italy, brand: Emilio Pucci, category: accessories, subcategory: sunglasses, season: spring/summer, color: yellow, bicolors: gold, pink, gender: women* |
| *Lanvin has created a new style of sunglasses with the new style of sunglasses with the new style of sunglasses with the new style of sunglasses with the new style of sunglasses with the new style of sunglasses with the new style of sunglasses with the new style of sunglasses with the new style of sunglasses with the n* |
| *Dior's Dior Homme sunglasses are crafted in Italy from lightweight acetate and feature a striking, oversized, oversized, rectangular shape.* |
| *A classic style from Fay, a perfect match for the spring/summer season. The brand, which is a member of the Italian Fashion Group, has been a reference in the fashion world for more than 50 years. This brand is characterized by the fact that it does not hesitate to experiment with new ideas and concepts, as well as with the most innovative fabrics and techniques. This is what makes Fay a brand that is constantly renewed and that is always able to find the right balance between tradition and innovation.* |

**Table 4.1:** Randomly chosen examples of descriptions generated by GPT-J without fine-tuning

| **Acceptable Generated Descriptions** |
|---|
| *This Dsquared2 sunglasses are made of high quality materials, it is a must-have for every man. It is a great gift for your boyfriend or husband.* |
| *The Carrera sunglasses are the perfect accessory for the fashion-conscious man.* |
| *Tory Burch's iconic designer bags have become a favorite of celebrities and everyday fashionistas alike.* |
| *Tory Burch has been creating iconic, covetable handbags since the company was founded in 1994. With a sophisticated, feminine aesthetic, Tory Burch bags are designed to be the perfect companion for any occasion.* |
| *the best of both worlds, the first sneakers to combine the benefits of a running shoe with the comfort of a casual shoe. The result is a sneaker that will not only make you run faster, but also make you feel better while you're doing it.* |

**Table 4.2:** Some examples of impressive description generated by GPT-J without fine-tuning

# Chapter 5

# Using GPT-Neo

## 5.1 About GPT-Neo

GPT-Neo, another model developed by EleutherAI before GPT-J, was used as an alternative to it. GPT-Neo was released in two versions, one with 1.3B parameters and one with 2.7B. Similar to GPT-J, GPT-Neo is trained on the Pile, and mostly on English data and the 1.3B version is comparable to GPT3-XL.

In experiment [10], the two versions of GPT-Neo were compared to GPT-3 models in title generation. The two different sizes of the model seemed to be working similarly in text generation. Moreover, GPT-Neo made generalizations similar to GPT-J and not always produced good enough text.

The most important advantage of GPT-Neo-1.3B was that it could be fine-tuned on Google Colab, using Google Cloud storage, which was an affordable option considering Colab TPUs are free and cloud storage is quite cheap.

## 5.2 Fine-tuning GPT-Neo

### 5.2.1 First Attempt

The problem with fine-tuning GPT-Neo was that not many labeled examples were available, i.e. not many good, interesting and descriptive product descriptions were available. The good product descriptions had to be close to the acceptable examples generated by GPT-J.

Since no sources with good product descriptions were available, the model was fine-tuned with acceptable results generated by GPT-J, similar to what was done in [11]. GPT-Neo-1.3B was fine-tuned with Colab TPUs using the manually chosen acceptable descriptions generated by GPT-J.

The results were very different from that of GPT-J. The number of missing generated product descriptions or descriptions that were about totally different categories of products, a description about a bag while the product is a hat, were noticeable. However the descriptions that were considered acceptable needed much less editing than before to become a presentable description, making the line between acceptable and non acceptable descriptions much more clear. There were also plenty of cases when one had to search for the generated product and extract in from a pile of nonsense generated by the model.

The "Grammar and Frequency" and "Coherence and Consistency" of 100 randomly chosen instances of the data were evaluated to be 3.09/5 and 2.40/5 respectively. Furthermore, about 53% of the results were acceptable. Acceptable results were the ones that had more than 2.5 points in average, and were in the form of a product description, i.e. results scoring perfect in "Grammar and Fluency" while not being a product description, thus getting a 1 in "Coherence and Consistency" weren't considered acceptable, despite scoring 3 in average.

### 5.2.2 Second Attempt

The second shot at fine-tuning Neo using labeled examples extracted from the company's only client with good product descriptions, Telablu, was not a success as the data set was too small and too specific to the limited products available on their website. Not to mention, the small data set contained lots of repetitive descriptions for different categories of products with little diversity in word choice or phrasing.

### 5.2.3 Third and Final Attempt

For the third fine-tuning attempt, data was extracted from 5 e-commerce websites including Flipkart, Amazon and EBay. A total of 2530 instances were chosen semi-manually from about 100K products, and the tags were transformed to the format of tags of Client-2's website. These descriptions combined with around 1200 instances generated by GPT-J were used as the training and validation sets for fine-tuning.

At this point, products with an average score greater than 3.5 were considered acceptable, as opposed to before when we were considering those with more than 2.5 points. By this new criteria, 83%, 464 out of 558, of the generated descriptions were acceptable. Moreover, the average "Grammar and Fluency" and "Coherence and Consistency" points of 100 randomly chosen instances generated descriptions were 4.47/5 and 3.86/5 respectively. The significant increase of "Grammar and Fluency" points could partially be explained by the fact that the model produced substantially less results that weren't considered a description at all.

A common problem in automatic text generation is that the system tends to generate safe answers without enough diversity [12]. And this phenomenon could be verified in our experiment. The final fine-tuned model with the best overall performance tends to generate less "creative" descriptions than our baseline which was GPT-J without fine-tuning. Thus a potential way forward could be to fine-tune the model such that lexical diversity is not sacrificed for accuracy and learn a model able to generate diversified texts. This is also crucial as human-generated texts are usually full of diversified contents, and in order to compete with manually-written descriptions, the generated descriptions need to be more diverse.

## 5.3 Inference with GPT-Neo

Inference with GPT-Neo was extremely slow and it took around 5 minutes to generate each result because of how the code was written. The authors had stated that their code wasn't efficient for sampling and had suggested to use HuggingFace and a GPU to infer using the model after training. There was no way to control the length of the generated output which had a significant effect on, inference time.

Moreover, a memory leak in one of TensorFlow's functions caused the session to run of of memory after around 40 results were generated and the run time had to be restarted.

### 5.3.1 Memory Leak

**Generating 6 descriptions consumed 1.21 GB of RAM never to be freed, and deleting all the references to all variables and data sets, calling the garbage collector and clearing the TF graph after each generation only got it down to 1.15 GB. Therefore, `objgraph` module was used to verify the existence of a memory leak.**

**The function `Estimator.predict()` was found to be the source of the memory leak. After each call to the function, several lists, tuples and dictionaries were created that were never freed, even after deleting all references to used variables and calling Python's garbage collector. The number of lists and dictionaries in the memory after each call to the function is portrayed in Table 5.1.**

| | **Before Calling** | **After 1st Call** | **After 3rd Call** | **After 6th Call** |
|---|---|---|---|---|
| **list** | 20939 | 134525 | 361720 | 702496 |
| **dict** | 64006 | 134741 | 276209 | 488439 |

**Table 5.1:** The number of lists and dictionaries in the memory after calling Estimator.predict()

### 5.3.2 Inference with Colab

**As Google Colab is meant to be used specifically for interactive use, automating the process could only be done to a certain degree. Certain tasks like connecting to another session when the current session runs out of ram and logging into Google Cloud and Drive should be carried out by a user. However, some methods such as using an auto click script, staying logged into google cloud and drive and automatically closing the current session and opening a new one when the session in closed due to exhaustion of memory, could also be investigated if needed.**

**Inference is done using one or multiple unlabeled .xlsx files containing product tags, like a normal catalog export.Similarly, training is done using one or multiple labeled .xlsx files containing the tags and the product description. Each row of the output file contains the URL of the product, the tags used to generate the description, the generated description and the list-like description, if present in the input files.**

The code for fine-tuning the model, pre-processing the data, generating descriptions and showing the output was structured into four .ipynb note-books to be used by a human, interactively. The four notebooks and their functions are as follows.

1. train-neo.ipynb:
    - The note-book is used to train the model using a batch of excel files with labeled data.
    - Data is read from all excel files in a directory, certain data cleaning operations are performed in it in order to keep only the desired instances and the result is written as a series of .txt files.
    - By default, all the data is written as the training set but there's also the option to write the data as train, validate or train, validate, test sets with any fraction.
    - The .txt files are then concatenated into a single large .txt file.
    - Then the code from GPT-Neo is used to tokenize the data set, upload the pre-trained model to the cloud storage and fine-tune it.
2. prepare.ipynb:
    - The note-book is used to transfer a batch of .xlsx files into data used by the following notebooks.
    - All the .xlsx files in a directory are read, the data is prepossessed by extracting required tags from the list-like description and cleaned, and the results are written as the unlabeled data containing only the tags which is used in the generation phase.
    - The corresponding labeled data containing the tags, the list-like description and the product URL is also written to another set of .txt files, the reference set, which is used in the result-showing phase.
    - A new checkpoint file is made at this point which is used in the generation notebook.
    - There is the option to set a limit for the instances read from each .xlsx file. this could be useful when trying to sample a few instances from many catalogs to ensure having a sparse and general test set. when used, all the instances of data in catalogs are read and a number of instances equal to the limit

is randomly selected, if the size of the catalog is larger than the limit.

3. generate-neo.ipynb:

   - The note-book uses code form GPT-Neo to generate product descriptions using the fine-tuned model for a batch of .txt inputs, the test set produced in the preceding notebook.
   - Results are saved to google drive, and products for which a description has been generated, are inserted in the checkpoint file, making it possible to resume the process in case the notebook runs out of memory and the session is closed.

4. show-results.ipynb:

   - The note-book is used to generate the output file, containing the product URL, the tags used for description generation, the generated description and the list-like description, if available.
   - The labeled references and the generated outputs of the previous notebooks are merged to make the final output file.

## 5.4 Multiple Descriptions

One of the major issues when presenting a single product (SKU) over numerous re-seller websites in a drop-shipping model is the fact that images and description are the same in all re-seller websites. This has a significant negative impact on SEO since Google and other search engines penalise sites with copied contents.

For this reason, the possibility of generating multiple product descriptions for a single product, i.e. a single set of tags, is of interest, as generating several descriptions for an SKU, enables the B2B e-commerce clients of the company to sell different product descriptions to each re-seller who wants unique descriptions.

For a given set of tags, eight different generated product descriptions are listed in Table 5.2

| **Tags:** | **subcategory: Sweater, gender: Women, season: Autumn winter, color: White** |
|---|---|
| **description 1** | *This sweater is a perfect addition to your wardrobe, and it is perfect for casual and professional occasions.* |
| **description 2** | *This sweater is a perfect combination of light and airy. It has a round neckline and a square pattern. It is perfect for a casual look.* |
| **description 3** | *This sweater is a perfect combination of soft and warm. It is made of a soft wool blend and features a hood with a button. It is a very comfortable and stylish sweater.* |
| **description 4** | *This sweater is a perfect addition to your wardrobe, and it is perfect for casual wear. This sweater is made of wool, and it is a very light and comfortable sweater. You can wear it with a pair of jeans or a pair of trousers.* |
| **description 5** | *Women's sweater with a round neckline and a sharp collar. It is made of wool and it is a must-have for the autumn season.* |
| **description 6** | *This sweater is a perfect addition to your wardrobe, and it is perfect for casual wear. This sweater is made of 100% wool, and it is a very light and comfortable sweater. You can wear it with a pair of jeans or a pair of trousers.* |
| **description 7** | *This sweater is a perfect combination of light and casual. It has a round neckline and a square hem. It is made of a light wool and it is very soft.* |
| **description 8** | *This sweater is a perfect combination of light and warm. It has a simple design and is made of a very soft and comfortable material. This sweater is a perfect piece for the cold winter days.* |

**Table 5.2:** Eight different generated product descriptions for a single product

## 5.5 Evaluation

**As the next step, the integrity of the descriptions was checked by a client. New products that had never been deployed on the website were requested from Client-2 to test the description generation process.**

The tags needed to be exhaustive for the model to work properly as seen in previous experiments but unfortunately this requirement wasn't met by the client and we were provided with products with very poor tags and no way to improve them. The tags consisted of category, color, gender and only about 15 percent of the products had tags with more information such as material, neckline, sleeve length and pattern.

Despite the poor quality of the tags, 80% of the generated descriptions, scored more than 3.5 and were acceptable and accurate to the limited provided information. "Grammar and Fluency" and "Coherence and Consistency" were 4.675/5 and 3.575/5 in average respectively. The points in the two areas were less than the results generated with quality product tags, which was expected considering the dependence of the process to having good tags. However, the points were still pretty high regardless of the poor tags. The client was satisfied with and moved by the results as well.

## 5.6 Human Supervision

Despite us reaching a point, where an acceptable percentage of the results were of high quality, due to the results being non-deterministic, there is no guarantee that the generated results won't be inaccurate, unrelated to the topic or even offensive at times. Therefore, human supervision is always essential. A number of particularly bad generated descriptions are listed in Table 5.3, corroborating the necessity of human supervision.

| **Generated Description** | **Reason** |
|---|---|
| *inhibitor of cell-mediated and humoral responses, as well as of the inflammatory and immunological effects of activation of cell-mediated and humoral immunity.* | Unrelated |
| *Jesus is the Messiah, the Son of the sixties, the one who will make all men of the nations thanks to his miracles.* | Unrelated |
| *enjoy your favorite foods and drinks with this pair of pants. The pants are made of a soft and comfortable fabric, and a the design is designed to make it easy for you to wear.* | Illogical |
| *4-in-1 belt with cardigan,leggings and shoes.* | Illogical |
| *humans are the worlds first to wear a leather boot,and we have a long way to go. from the classic black and white to the color black and silver, this pair of boots is perfect for the season.* | Illogical |
| *migraine-controlling pants with front and back pockets.* | Illogical |
| *It is a perfect shirt for a basic woman's wardrobe.* | Offensive |
| *The Calvin Klein Bag is a small, versatile bag that can be worn as a jacket or a casual bag.* | Illogical |
| *A simple, stylish and casual skirt for the summer. Made of a soft cotton fabric, it has a round neckline and a V-neck* | Illogical |

**Table 5.3:** Particularly bad generated descriptions corroborating the necessity of human supervision

# Chapter 6

# Better Results

In this chapter, some possible future directions for fine-tuning a better model are discussed.

## 6.1 Reinforcement Learning

Good product descriptions are characterized by not only being fluent, accurate and appealing, but also having high SEO scores. Therefore, another way to check the integrity of the generated product descriptions could be to check how many SEO keywords they contain. A possible solution to improve the generated product descriptions in terms of containing SEO keywords, could be to exploit reinforcement learning.

Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment [13]. Through the years, Reinforcement Learning has been used in many different domains and has experienced great success and impressive results. Some of these domains are playing board games such as AlphaGo Zero [14], robot navigation [15], motion analysis and visual tracking [16] and financial asset trading [17].

In reinforcement learning, the reward function is an indication of how good or bad a given state is, and the goal of the agent is to maximize the reward. In our case, the PTM would be fine-tuned using RL, the reward function being the number of SEO keywords used in the generated product description. The agent will try to maximize this number which will lead to better generated descriptions in terms of SEO keywords.

A similar work to this suggestion, has been done in [18] using RL to refine the RNN sequence prediction model and produce more globally structured and harmonious music. The reward function being a combination of rewards based on rules of music theory and the output of another trained RNN, The reward RNN. This work introduces a novel method to enhance the previous works of generating music with Deep learning, e.g. [19] by using reinforcement learning.

To give more structure to the produced melodies, reinforcement learning with music theory laws is used in [18]. The reward function is a combination of music theory rules which gives the melody a structure and the reward RNN which is necessary for the model to remain “creative”, rather than learning a simple composition that can easily exploit rewards corresponding to music theory. The latter could also serve our purpose as one of the primary required characteristics of the description is being creative enough to be attractive for the potential customer, as previously mentioned.

## 6.2 Human Interaction

Another idea is to exploit interactive reinforcement learning by introducing a human advisor to grade the generated product descriptions. This could help improve the aspects of the generated descriptions that are important for us, and make the descriptions match our preferences. In a similar work by OpenAI [20], human advisors were used to fine-tune GPT-2 for various downstream tasks. Although this approach could potentially improve the quality of generated description by far, considering its downsides, it might not be worthwhile.

First of all, the process is very time-consuming. Moreover, training the advisors to grade the descriptions based on a specific set of rules is expensive, and regardless of how well they have been synchronized, there is always the risk of having a bias in the way advisors teach the model. The style of teaching could also change over time as the advisor is exposed to results generated by the agent.

As the experiment goes on, advisors might get habituated to the generated results, or the results might influence their couching style. In the experiment done in [21], the advisors gave more detailed feedback in the beginning, but as the model became sufficiently good, the feedback

**dropped, occasional mistakes were overlooked, and no positive feedback was given for desirable behavior.**

# Chapter 7

# Conclusion and Future Work

## 7.1 Conclusion

**Experiments done using GPT-J and GPT-Neo, aimed at automating the process of generating product descriptions, lead to more than 80% of the generated descriptions being sufficiently fluent, almost accurate and appealing. A summary of the quality of the results obtained by different models, is listed in Table 7.1.**

| **Model** | **Fine-tuned with** | **Grammar** | **Coherence** | **Acceptable** |
|---|---|---|---|---|
| GPT-J | no fine-tuning | 3.96 | 2.49 | 43% |
| GPT-Neo | acceptable results of GPT-J | 3.09 | 2.40 | 53% |
| GPT-Neo | good descriptions from 5 e-commerce websites and the acceptable results of GPT-J | 4.47 | 3.86 | 83% |

**Table 7.1:** A summary of the quality of the generated results, the "Grammar" and "Coherence" points refer to "Grammar and Fluency" and "Coherence and Consistency" points respectively.

**However, much obstacles still remain including the need to train and run the model without having to use Google Colab or paying the hefty prices of Google Cloud TPUs, not having the possibility to generate**

**descriptions in any language other than English, and being dependent on the quality of the provided product tags.**

## 7.2 Future Work

**The two primary future directions of this line of experiments will be to find a way to use the model in production, either by purchasing a GPU or using Cloud TPUs, and to translate the generated product descriptions to other languages, using machine translation.**

**For training and inference on GPU, a version of GPT-Neo, called GPT-Neox, is available which is compatible with HuggingFace, supports different normalization methods, as well as different optimizers. Moreover, GPT-Neox provides support of WandB to records and visualize experiments as well as TensorBoard. At the time this report was written, they were working on adding the feature to distill the model which could be useful in the future as well.**

**Future projects continuing this work could be focused on generating descriptions from the product pictures using computer vision and a text generation model, making the process reliable with no dependence on having good tags, or generating recommended products from users input in the form of text description using NLU and semantic search.**

# Appendix A

# Peculiar Results

| **Peculiar Generated Descriptions** |
|---|
| *Armani Jeans Shirt made in Tunisia, are you looking for a brand new Armani Jeans Shirt? You are in the right place!* |
| *This is a small accessory for the little girls. This cute little bag is made of soft and durable material and has a nice zippered closure. This bag is ideal for the little girls to keep their little things in.* |
| *This coat is so chic, it's so comfortable, and it's so warm! It's made of a soft wool blend that's so soft, it feels like you're wearing a hug.* |
| *These pants are made from a stretchy fabric that will hug your curves in all the right places. The fabric is also durable and easy to clean. This pair of pants is designed to fit your lifestyle and your style.* |
| *This winter we are all about the turtleneck. It's a piece that you can wear all year round, thanks to the fact that it can be worn under a coat or over a sweater. This time we have opted for a grey version.* |
| *It's time to be bold and make a statement. With the Adidas Y-3 Yohji Yamamoto Sweatshirt, you can.* |
| *A slipper that fits like a glove.* |

**Table A.1:** Some examples of peculiar generated descriptions

**Some Peculiar examples of generated product descriptions are listed in Table A.1.**

# Bibliography

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: `10.18653/v1/N19-1423`. URL: `https://aclanthology.org/N19-1423` (cit. on p. 1).

[2] Tom B. Brown et al. «Language Models are Few-Shot Learners». In: *CoRR* abs/2005.14165 (2020). arXiv: `2005.14165`. URL: `https://arxiv.org/abs/2005.14165` (cit. on pp. 1, 2).

[3] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. 3rd edition. Pearson, Harlow, 2010 (cit. on p. 1).

[4] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. «Pre-trained Models for Natural Language Processing: A Survey». In: *CoRR* abs/2003.08271 (2020). arXiv: `2003.08271`. URL: `https://arxiv.org/abs/2003.08271` (cit. on p. 2).

[5] Jared Kaplan et al. «Scaling Laws for Neural Language Models». In: *CoRR* abs/2001.08361 (2020). arXiv: `2001.08361`. URL: `https://arxiv.org/abs/2001.08361` (cit. on p. 2).

[6] Alec Radford and Karthik Narasimhan. «Improving Language Understanding by Generative Pre-Training». In: 2018 (cit. on p. 2).

[7] William Fedus, Barret Zoph, and Noam Shazeer. «Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity». In: *CoRR* abs/2101.03961 (2021). arXiv: `2101.03961`. URL: `https://arxiv.org/abs/2101.03961` (cit. on p. 2).

[8] Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. «Paraphrase Generation with Deep Reinforcement Learning». In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3865–3878. DOI:

10.18653/v1/D18-1421. URL: https://aclanthology.org/D18-1421 (cit. on p. 5).

[9] Mitchell A. Gordon, Kevin Duh, and Nicholas Andrews. «Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning». In: *CoRR* abs/2002.08307 (2020). arXiv: 2002.08307. URL: https://arxiv.org/abs/2002.08307 (cit. on p. 8).

[10] Peng Z. and Budhkar A. *GPT-Neo vs. GPT-3: Are Commercialized NLP Models Realy That Much Better?* https://medium.com/georgian-impact-blog/gpt-neo-vs-gpt-3-are-commercialized-nlp-models-really-that-much-better-f4c73ffce10b. 2021 (cit. on p. 13).

[11] Alberto Poncelas and Andy Way. «Selecting Artificially-Generated Sentences for Fine-Tuning Neural Machine Translation». In: *Proceedings of the 12th International Conference on Natural Language Generation.* Tokyo, Japan: Association for Computational Linguistics, Oct. 2019, pp. 219–228. DOI: 10.18653/v1/W19-8629. URL: https://aclanthology.org/W19-8629 (cit. on p. 14).

[12] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. «Deep Reinforcement Learning for Dialogue Generation». In: *CoRR* abs/1606.01541 (2016). arXiv: 1606.01541. URL: http://arxiv.org/abs/1606.01541 (cit. on p. 15).

[13] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. «Reinforcement Learning: A Survey». In: *CoRR* cs.AI/9605103 (1996). URL: https://arxiv.org/abs/cs/9605103 (cit. on p. 22).

[14] David Silver et al. «Mastering the game of Go without human knowledge». In: *Nature* 550 (Oct. 2017), pp. 354–. URL: http://dx.doi.org/10.1038/nature24270 (cit. on p. 22).

[15] Piotr Mirowski et al. «Learning to Navigate in Complex Environments». In: *CoRR* abs/1611.03673 (2016). arXiv: 1611.03673. URL: http://arxiv.org/abs/1611.03673 (cit. on p. 22).

[16] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. «Action-Decision Networks for Visual Tracking With Deep Reinforcement Learning». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* July 2017 (cit. on p. 22).

[17] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. «Deep Direct Reinforcement Learning for Financial Signal Representation and Trading». In: *IEEE Transactions on Neural Networks and Learning Systems* 28 (Feb. 2016), pp. 1–12. DOI: 10.1109/TNNLS.2016.2522401 (cit. on p. 22).

[18] Natasha Jaques, Shixiang Gu, Richard E. Turner, and Douglas Eck. «Generating Music by Fine-Tuning Recurrent Neural Networks with Reinforcement Learning». In: *Deep Reinforcement Learning Workshop, NIPS*. 2016 (cit. on p. 23).

[19] Douglas Eck and Jürgen Schmidhuber. «Finding temporal structure in music: Blues improvisation with LSTM recurrent networks». In: vol. 12. Feb. 2002, pp. 747–756. ISBN: 0-7803-7616-1. DOI: `10.1109/NNSP.2002.1030094` (cit. on p. 23).

[20] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. «Fine-Tuning Language Models from Human Preferences». In: *CoRR* abs/1909.08593 (2019). arXiv: `1909.08593`. URL: `http://arxiv.org/abs/1909.08593` (cit. on p. 23).

[21] Andrea Thomaz, Guy Hoffman, and Cynthia Breazeal. «C.: Real-time interactive reinforcement learning for robots». In: (Dec. 2008) (cit. on p. 23).