BT5153 Group Project Report
Product Description Generator from Images for Online Fashion Retailers

Group 03:
Chong Zhi Ting (A0103510E) | Hiroyuki Tsujikami (A0218895L) | Juan Huang (A0218928R)
Liu Siqing (A0127160R) | Suen Ao Xiang (A0113842M)

Abstract

Small fashion apparel businesses have increasingly leveraged e-commerce platforms to market and sell their clothing products. As these businesses might be keen to enter global markets, it is important that they are able to market and describe their products effectively to ensure customer confidence. In this paper, we thus explore various deep learning methods to develop an automated product description generator model which can be implemented on e-commerce platforms and used by these businesses to curate descriptions for their products at a fraction of the time and cost. We have achieved promising results with the Transformer model and will discuss further improvements that could enhance our model's performance.

1. Introduction

The advancement of technology and increasing usage and access to mobile devices has led to an increase in the worldwide adoption of e-commerce in the past decade. Mega online retail companies such as Amazon and Alibaba have seen a huge growth in their businesses over the years (Holmes, 2019; Zeng, 2018). The outbreak of the Covid-19 pandemic has further accelerated the adoption of e-commerce worldwide, where consumers have reported a large shift towards online compared to physical in-store purchases, according to a McKinsey survey (Charm et al., 2020). Given the shift towards online retail, it is important that firms are able to engage with their customers through effective marketing of their products on e-commerce platforms, in order to enhance customer experience and ensure business sustainability.

In particular, small fashion apparel businesses have increasingly leveraged e-commerce platforms to market and sell their products to a wider range of consumers globally. As these businesses might be keen to enter new consumer markets where English is primarily used and/or understood (e.g. USA, Europe, Southeast Asia), it would be important that they are able to market their products well with not just good product photography, but also effective descriptions in their product listings, in order to ensure customer confidence in their products. The writing of product descriptions in English might however be challenging and time consuming for small businesses that have many apparel items in their product range, or where English is not their primary business language.

As such, in our project, we aim to develop a product description generator in English for fashion apparel products that could be implemented on e-commerce platforms to help small fashion apparel businesses better market their items online to predominantly English-speaking markets.

2. Project Objective

Our main objective of the project is to apply deep learning image captioning techniques to develop a fashion apparel product description model that receives an image of a clothing item and outputs accurate captions describing the product, based on its image. To do so, we require a dataset on the images of fashion items and their accompanying product descriptions in English. Given that production descriptions and vocabulary used for fashion items (e.g. clothing, bags, footwear, accessories) vary widely depending on their functional purpose and category, we have decided to focus our project on clothing items only.

In the following sections of this report, we first outline the potential outcomes and business applications of our project, followed by details of our dataset, and the methodology we have adopted for our project.

3. Business Applications

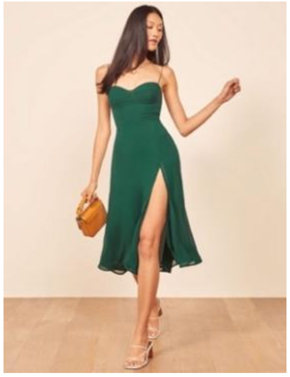
An application for an intelligent product description generator would be for e-commerce fashion retailers to curate quality descriptions for their product catalogues at a fraction of the time and cost. Particularly for brands originating from non-English speaking countries such as China, Korea and Japan, whose fashion styles are gaining considerable traction in English speaking countries in the west. Curating thousands of attractive product descriptions could turn out to be time-consuming and expensive, on top of that, a poorly worded, or incorrect product description would have an adverse impact on their conversion rate. By curating unique descriptions specific to their products, e-commerce retailers are also able to improve their search engine optimization (SEO) performance, where uniqueness and relevance would be essential to reach the right customers, all of which translates to better conversion rates.

An additional feature for our solution would be in the context of keyword search. Since image captioning techniques generate unique captions for individual images, online fashion retailers could leverage on this solution to improve on their website's search function, where users could describe the product they are interested in, instead of relying on traditional filter tags such as product type, colour and brands. While the scope of this project focuses on clothing items, a potential extension of this application would be to other fashion items such as footwear and accessories.

4. Dataset

4.1 Data Sources and Collection

Image captioning applies deep learning algorithms in Natural Language Processing (NLP) and Computer Vision, which necessitates large datasets for model training. To obtain such big datasets, we implemented web scrapping on various popular fashion brands' online shopping websites to retrieve relevant clothing product data. Data was collected from a range of fashion brands including H&M, Adidas, Uniqlo and Vero Moda, to ensure variety and better generalization of our models. Furthermore, we also collected data for both female and male, and across diverse clothing categories. For our web scrapping tools, we mainly used a third-party scrapping software named Octoparse, as well as Selenium, a web-scraper package in Python. An example of a clothing product image and its description is illustrated in **Figure 1**.



Product Description:

"Somebody's fancy. Shop the Juliette midi dress with a sweetheart neckline, tie straps and a side slit. Slim fitting throughout the bodice."

Figure 1: Sample Product and Description

4.2 Data Overview

We have scrapped a total of 16 popular fashion brands' clothing product data, which amounts to 51,109 clothing items, and compiled the data from different brands into a single large dataset. The data fields available for each item in our dataset are summarized in **Table 1**.

COLUMN NAME	DATA TYPE	VARIABLE TYPE	DESCRIPTION
TITLE	Text	Independent	Product's name
URL	Link	Independent	Link to the product's image
DESCRIPTION	Text	Dependent	Product description provided on the clothing retailers' website
SEX	Text	Independent	Gender category for the product
MATERIAL	Text	Independent	Material of the product
PRICE	Text	Independent	Product's price
BRAND	Text	Independent	Fashion brand of the product
COLOUR	Text	Independent	Product's colour

Table 1: Summary of Raw Dataset

4.3 Data Exploration and Cleaning

Following data collection, we performed cleaning and exploratory data analysis on our dataset. **Figure 2** shows a word cloud generated from our description dataset. Interestingly, "size s", "t shirt" and "model" seem to occur frequently. This suggests that t-shirt categories are common across brands, which is expected since it is one of the most basic clothing items.

Also, we investigated the frequently occurring word "you[][]e" and found that there are some Chinese characters present, such as "窶" (30,249 counts) and "决" (11,897 counts). Moreover, "窶" often occur in descriptions with the word "you 窶况 re", which should have been "you're" with an apostrophe. This is likely due to the limitation of the scraping packages and character formatting, resulting in many irrelevant characters scraped. Thus, we replaced these Chinese characters with their actual notation in our dataset.

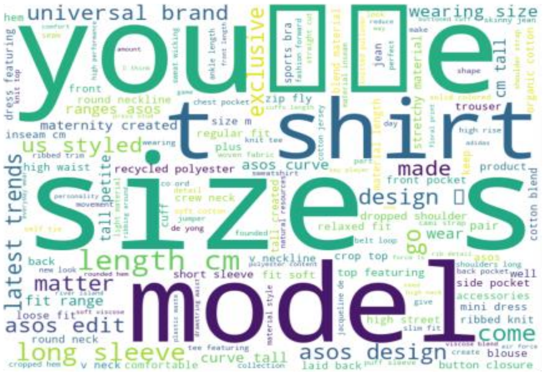


Figure 2: Product Descriptions Word Cloud (Before Processing)

Subsequently, we performed further data pre-processing on our descriptions dataset by removing punctuations and stand-alone numbers, as these tend to be less relevant in product descriptions. We also replaced English contractions with their longer form and converted the text to lower case.

While exploring the data, we observed that some product descriptions included the name of the brand. To standardize across brands, we thus replaced the name of the brand, such as “Nike”, to “brand”. In addition, some brands had the same descriptions for different product images due to colour variation. We thus removed duplicate items having the exact same product description but different image.

Finally, the style of the description for some brands was found to be completely different from other brands. For instance, an example of a clothing description for ASOS is “This is ASOS go-to for all the latest trends, no matter who you are, where from and what up to. Exclusive to ASOS, our universal brand is here for you, and comes in Plus and Tall. Created by us, styled by you.” Despite being a description for a hoodie, the text seems to be describing the brand ASOS rather than the product itself. Furthermore, some product descriptions consist of short words in bullet points, such as “- 4-pockets - Closure at front - Length: 72 cm in size S/34”, instead of a full description. We removed these data from the dataset given that their product descriptions were less useful and relevant in describing clothing items.

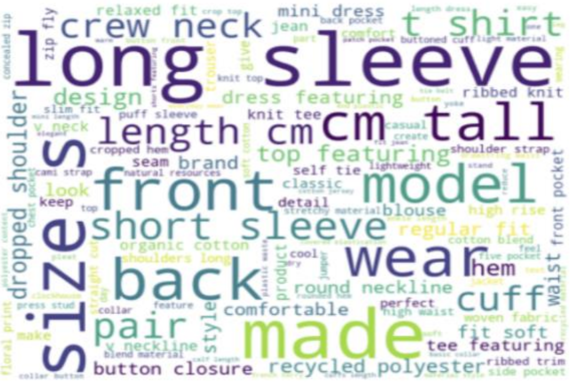


Figure 3: Product Descriptions Word Cloud (After Processing)

After data pre-processing, we have 18,680 product images and descriptions in our final dataset for model development. **Figure 3** shows the word cloud for our final dataset.

4.4 Preliminary Exploratory Data Analysis (EDA)

We carried out preliminary EDA to obtain an overview and initial insights from our final dataset. As shown in **Figure 4**, we observed that our dataset has a higher representation of women’s clothing than men’s, as the amount of data for men’s is less than half of that for women’s. Pertaining to retailer brands, the three prominent brands, namely Forever21, H&M, and C&A also have distinguishably more data compared to the rest. These patterns are also observed in the industry where there are typically more women’s apparel products than men’s, and large retailers have a much wider product range than smaller fashion boutiques or sports brands. Hence, our model might generalize and perform better on female products with wordings reflecting the writing styles of the larger fashion retail brands more so than the smaller ones.

Furthermore, we observe that the average description length of products for men is slightly longer than that for women. Sports brands (i.e. Adidas, Nike) also tend to have longer product descriptions than other retail brands, as they typically elaborate more on their clothing products’ performances for sports use.

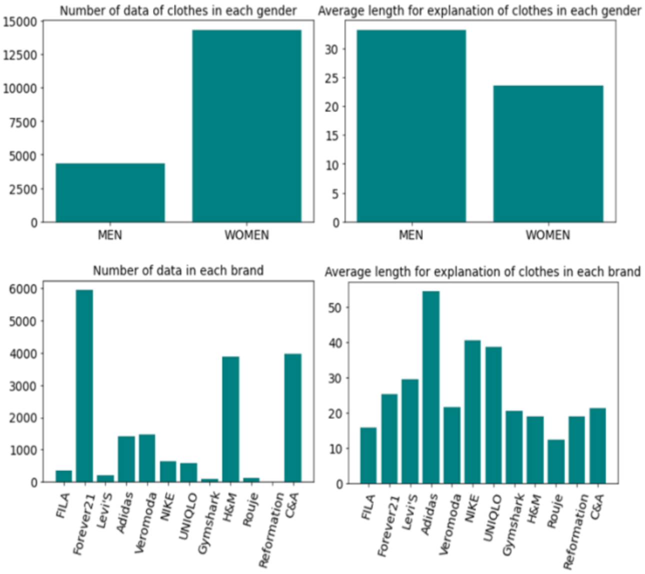


Figure 4: Preliminary EDA

5. Methodology

The generation of descriptions from clothing images involves the application of deep learning models with image and text data as inputs. In this section, we discuss the data pre-processing methods, neural networks, and search algorithms that have used to develop our product descriptor model for the project.

5.1 Data Pre-processing

5.1.1 Image Data

Prior to the training of our product descriptor model, we first processed the clothing images in our dataset into feature vectors. We adopted the transfer learning approach by using an established pre-trained image recognition model for feature extraction from images, namely InceptionV3 (Szegedy et al., 2016), as illustrated in an example of our model structure in **Figure 6**. InceptionV3 is a model widely used for image recognition tasks, built on a convolutional neural network trained on the ImageNet database which consists of over 14 million images, and known for its proven accuracy. Given our smaller data size of 18,680 images, using a pre-trained model to extract feature vectors from images could allow for better generalizability of our model, as well as reducing the time required for model training. The extracted image features of dimension 2,048 are subsequently used as inputs to our product description generator models.

5.1.2 Product Description Data

For our product descriptions dataset, we performed data pre-processing by transforming the textual descriptions into scalar vectors. The transformed vectors were subsequently used as inputs to a word embedding layer, where we adopted a transfer learning approach by utilising pre-trained word vectors of dimension 50 from the GloVe model (Pennington et al., 2014). This model was trained on corpus data from Wikipedia and Gigaword 5, and provides pre-trained vectors for a vocabulary size of 400,000 words. By using the pre-trained vectors from the GloVe model, we are able to obtain vector representations of words in our dataset that reflect the semantic similarity between words, which will likely be useful for our task. The usage of GloVe is also depicted in our model illustration in **Figure 6**. Our dataset has a vocabulary size of 8,922 unique words, out of which pre-trained GloVe word vectors were available for 8,362 of them.

In addition to using pre-trained word vectors, we also considered training a word embedding layer from scratch in one of our candidate models, which we will further discuss in Section 5.2.2.

As part of the word embedding process, we have pre-defined a maximum input text length of 62 words. Given that the product descriptions in our dataset are generally shorter with a mean length of 31 words and a right-skewed distribution as shown in **Figure 5** a sequence length of 62 would account for most descriptions and thus be a reasonable cut-off. Taking into consideration that the performance of text generator models might deteriorate beyond a certain output text length, we have also set an output text length of 32 words, which we use for evaluation of model performance.

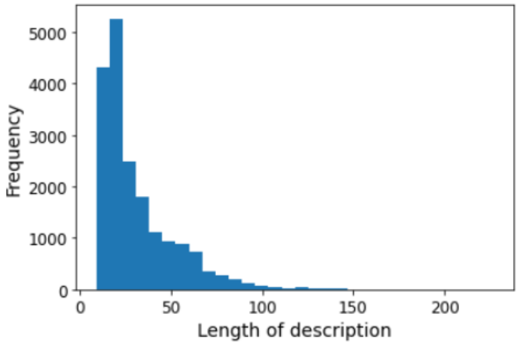
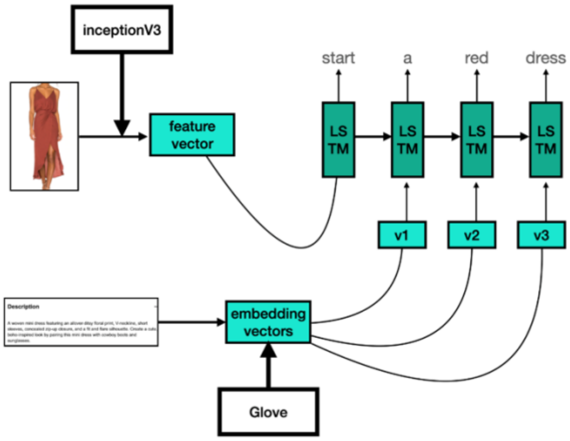


Figure 5: Distribution of description length

5.2 Product Image Descriptor Models

Following the data pre-processing steps in Section 5.1, the extracted image and product description feature data are fed into our product description generator models for training. In this project, we have considered two candidate models, namely long-short term memory networks, and transformers. In this section, we discuss the architecture that we have used for these models.

5.2.1 Long-Short Term Memory Networks (LSTM)



A woven mini dress featuring an allover ditsy floral print, V-neckline, short sleeves, concealed zip-up closure, and a fit and flare silhouette. Create a cute boho-inspired look by pairing this mini dress with cowboy boots and sunglasses.

Figure 6: Illustration of proposed LSTM model

Recurrent Neural Networks (RNN) are a class of neural network models that are widely used for NLP tasks. RNN models utilise previous outputs to be used for its hidden layers recurrently and thus enables the model to take into account historical information which is useful for predictions involving sequence data. In NLP applications, RNNs are commonly used for the prediction of the next word in a text sequence.

In a basic RNN model, a single tanh layer is usually repeated and thus the backpropagation gradients could sometimes converge to zero during the model training process. This makes it difficult for the model to grasp the long-term relationship and context of the data. As such, long-short term memory networks (LSTM) were introduced as an improved version of the basic RNN which enables the model to learn long-term dependencies. The LSTM model has four gates, namely the forget gate, output gate, input gate, and candidate gate, which selectively filter data in each process, thus avoiding the gradient vanishing problem of RNNs. This structure enables the model to better learn the context and long-term relationship between words in a sentence, and thus works better for most NLP tasks. **Figure 6** illustrates the use of LSTM for our product description generator model.

The overall architecture that we have adopted for our LSTM-based model is shown in **Figure 7**. There are two input layers, one that takes in the description vectors and passes them to the embedding layer followed by the LSTM layer, and another that takes in the pre-processed image feature vectors. Fixed-length vectors from both image and text processing layers are subsequently joined together and passed through an additional dense layer before the final softmax output layer. With this architecture, the model uses previous words and the product image to generate the next word in a sequence, allowing for the generation of product descriptions. We have used Adam for the model's optimization algorithm with a learning rate of 0.01 and batch size of 20.

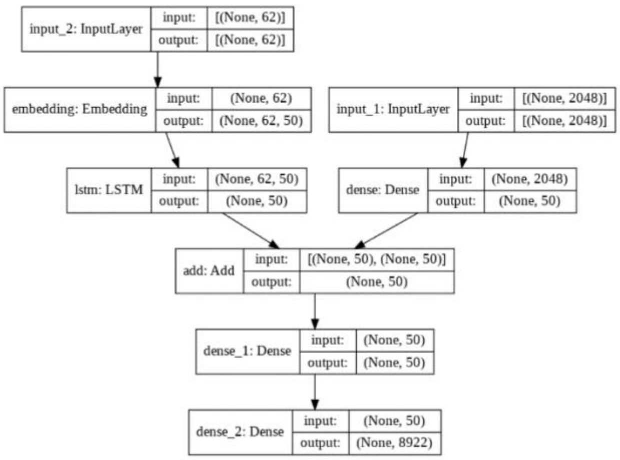


Figure 7: Architecture of LSTM Model

5.2.2 TRADITIONAL TRANSFORMER (TRANSFORMER 1)

Similar to the LSTM model, Transformer is also a sequence-to-sequence model that adopts an encoder-decoder architecture. However, unlike LSTM, Transformer enables parallelization in the processing of input sequences without time steps involved and incorporates various attention mechanisms. Encoder-decoder attention is one attention mechanism applied in the Transformer's decoder stack that finds a correlation between output variables and input variables, which enhances the model predictive accuracy. In addition, self-attention is another attention mechanism utilized in both encoder and decoder stacks to enable better representation encoded for each element in an input or output sequence, through relating its position to that of other elements in the same sequence. These attention mechanisms in Transformer gives it an edge over LSTM in terms of model performance.

Furthermore, another advancement in Transformer over LSTM is the enablement of parallelization. In Transformer's encoder, each element in the input sequence flows through its own path, and dependencies exist between different paths in the self-attention layer. However, such dependencies are absent in the feed-forward layer of the encoder, and hence various paths can be processed in parallel. This parallelization of input processing substantially boosts the Transformer's running speed as compared to LSTM where inputs can only be processed in sequence (Alammar, 2018).

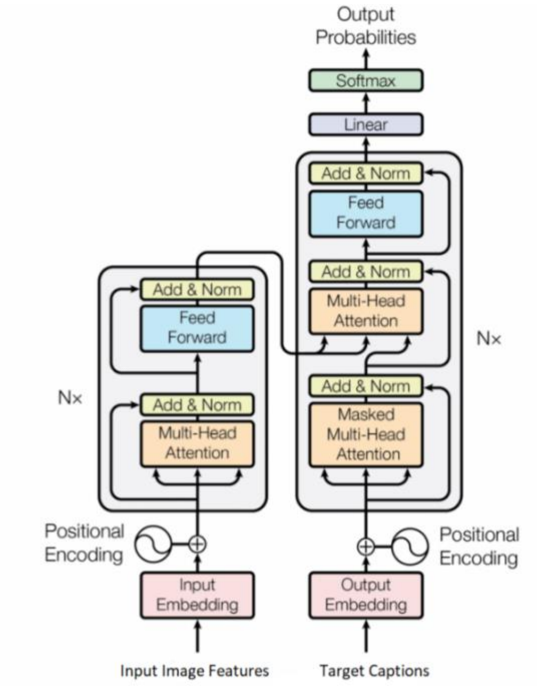


Figure 8: Transformer 1 Structure

In our project, we designed and experimented with two Transformer models with different architectures. The first Transformer model has a traditional architecture (Transformer 1) that was adapted from the model of an online author in his attempt at a similar image captioning task (Gautam, 2021).

For Transformer 1, an architecture of encoder-decoder layer with positional encoding and multi-head attention mechanism was constructed, as illustrated in **Figure 8**. In the encoder stack, features of training images extracted through InceptionV3 were embedded. In each encoder, there are two sub-layers which are the multi-head self-attention layer and position-wise fully connected feed-forward network. Layer normalization was applied to each sub-layer as well.

In the decoder stack, target caption sequences of training images are passed. Each caption sequence was pre-processed and tokenized via TensorFlow tokenizer, with a vocabulary size of 8922 and maximum input length of 62. On top of the two abovementioned sub-layers, another sub-layer of multi-head attention was added in each decoder to receive the encoder block's output to compute the correlation between input and output elements. Lastly, layer normalization is also included for each sub-layer in the decoder block.

For positional encoding in Transformer 1, sine and cosine functions with different frequencies are applied. For each input vector, the cosine function is used to create a vector for every odd index while the sine function is used to create a vector for each even index.

Lastly, various transformer model hyperparameters are also defined through our experimentations, as illustrated in **Table 2**.

HYPERPARAMETER	VALUE	REMARKS
NUM_LAYER	6	Number of layers
D_MODEL	50	Embedding dimension
DFF	2048	Hidden layer dimension
NUM_HEADS	5	Number of attention heads
ROW_SIZE	8	Row size
COL_SIZE	8	Column size
VOCAB_SIZE	8922	Vocabulary size
DROPOUT_RATE	0.1	Dropout rate

Table 2: Transformer 1 Model Hyperparameters

5.2.3 Modified Transformer (Transformer 2)

In addition to the traditional transformer architecture, our group also built another Transformer with a decoder-only architecture (Transformer 2) in comparison with Transformer 1. The rationale behind this is that the sequence and position information may not be relevant for images in contrast to their importance in a text sequence. Hence, Transformer 2's encoder does not apply a self-attention mechanism.

Moreover, another difference between the architecture of the two Transformers is that while Transformer 1 does not construct word embeddings using pre-trained models, Transformer 2 utilises GloVe pre-trained vectors for its word embeddings. Our hypothesis is that the pre-trained GloVe vectors would deliver a better quality of word embeddings which could, in turn, improve our model performance. With different architectures, we intend to examine and compare the two Transformers' performances with LSTM. The hyperparameters defined for Transformer 2 are shown in **Table 3**.

HYPERPARAMETER	VALUE	REMARKS
NUM_LAYER	6	Number of layers
D_MODEL	50	Embedding dimension
DFF	512	Hidden layer dimension
NUM_HEADS	5	Number of attention heads
MAX_POSITION _ENCODING	62	Number of positions encoded
VOCAB_SIZE	8922	Vocabulary size
DROPOUT_RATE	0.1	Dropout rate

Table 3: Transformer 2 Model Hyperparameters

5.3 Search Algorithms

The caption generation process for the various model architectures described above is similar, whereby the inputs contain an image and a text sequence. The process is initiated with a beginning-of-sentence token 'ss', which indicates the beginning of the sentence, and the model predicts one word token per iteration. The predicted word would then be concatenated with the original input text sequence, and to be used as input for the next prediction time step. This process is repeated until the model predicts the end-of-sentence token 'ee', or the caption generated reaches the maximum length, which we have indicated as 32 words.

5.3.1 Greedy Search

As the outputs from the different model architectures can be interpreted as the probabilities of the next word following the input sequence, a simple approach to prediction would be to select the word with the highest probability. This approach is regarded as the greedy search approach. However, this approach may not be ideal for predicting word sequences such as caption generation, as the greedy search approach does not guarantee that the joint probability of the predicted caption would be the highest.

Another issue with the greedy search approach is that some words may have consistently high probabilities if selected independently, thereby resulting in the predicted caption having repeating words, or in some cases, having ‘stuck’ with the same words for the remaining of the caption.

5.3.2 Beam Search

In order to mitigate the issues stemming from the greedy search approach, we utilised another search algorithm called beam search (Zhang et al., 2020). This approach stores a pre-determined number of candidate words *k*, and iteratively grows the tree structure for all candidate words. The branches of the tree are pruned based on the highest *k* joint probabilities at the particular time step.

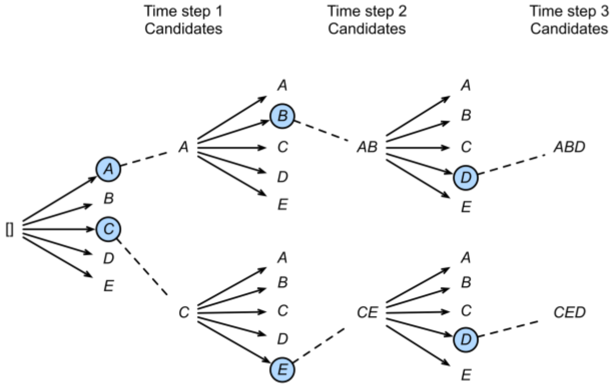
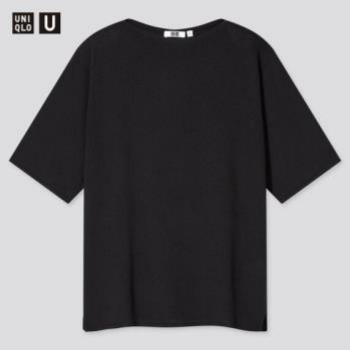


Figure 9: Illustration of Beam Search Approach (Adapted from d2l.ai)

In this case, *k* would be a hyperparameter to be tuned, where *k = 1* is analogous to greedy search, while *k = total word vocabulary* would be an exhaustive search approach where all possible joint probabilities are considered. Due to the iterative nature of beam search, increasing the *k* value would exponentially increase the computational resource required. Therefore, although beam search still does not guarantee an optimal solution with the best joint probabilities, having *k = 2* would sufficiently mitigate the issues present in the greedy search approach. **Figure 10** shows a comparison between both search algorithms.



Greedy Search:

“a knit tee featuring a front graphic of the text of the text of the text of the text of the text of the text of the text.”

Beam Search:

“a knit tee featuring a crew neck long sleeves and a crew neck.”

Figure 10: Comparison between Greedy Search and Beam Search Predictions

5.4 Evaluation Methodology

To ensure consistency in our model comparisons, the models were evaluated on the same validation set, which was 20% of the full dataset. We utilized four metrics to guide our decisions on selecting the best model, namely accuracy score, Bilingual Evaluation Understudy (BLEU) score, Recall-Oriented Understudy for Gisting Evaluation (ROUGE) and human evaluation of the generated captions.

5.4.1 Accuracy Score

Accuracy score is a straightforward method of calculating the percentage of words predicted correctly when compared to the actual caption. However, this naïve approach may not be a good representative for this project’s use-case as in natural language, the same information may be presented in different parts of the sentence.

5.4.2 Bilingual Evaluation Understudy (BLEU)

BLEU score is a popular method for evaluating generated sentences against the reference sentence. The score is calculated by counting the matching n-grams in the generated sentence to the n-grams in the reference text, where unigram is a comparison between tokens, and bigram is between word pairs. A perfect match would result in a score of 1.0, while a perfect mismatch would have a score of 0. A common approach to combining the various n-gram scores would be an equally weighted score of up to 4-gram. However, there are also areas where the BLEU score is lacking. For example, comparing the captions shown in **Figure 10**, the greedy search approach would have had a higher BLEU score than the beam search if the repeating word is present in the actual text, as this would have high precision. In addition to that, the beam search approach would also be penalized due to the shorter caption length.

5.4.3 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

ROUGE is a modification of BLEU that also considers recall rather than precision only. The F1 score, which is a weighted average of precision and recall, could be calculated as well. This is a better representative metric for the comparison of model performances for our project use-case, as it requires the model to capture as many relevant words as possible without inflating the scores of repetitive words and penalizing shorter captions.

As there are many variants of the ROUGE metric, we decided on ROUGE-L, which measures the longest common subsequence (LCS) between the generated caption and actual caption. The key idea is that a longer subsequence that is shared would indicate higher similarity between the generated and actual caption, and this also allows varying n-grams to be considered.

5.4.4 Human evaluation

Despite the various evaluation metrics described above, it is understandably complex to quantitatively measure the effectiveness of caption generators. Therefore, we also considered human evaluations in addition to the abovementioned metrics, such as conducting manual checks on whether the generated caption conveys the necessary information from the image. We also considered the uniqueness of generated captions in the validation dataset, whereby a higher number of unique captions would indicate that the model generalizes well.

6. Results

In this section, we report the performance results obtained from our methods described in Section 5, discuss the pros and cons of the various approaches, and examine the generated descriptions for several case studies.

6.1 Model Comparison

Table 4 summarizes the performance of our three models, LSTM, Transformer 1, and Transformer 2 on the validation set, first using the greedy search approach. Transformer 2 was found to result in the highest validation accuracy, followed by LSTM and Transformer 1. On the other hand, LSTM and Transformer 1 scored better on BLEU and ROUGE metrics than Transformer 2. As mentioned in Section 5.4.4, we further compared the models on their proportion of unique descriptions generated on the validation set. Transformer 2 was found to generate a much higher proportion of unique descriptions (63%) as compared to LSTM (2%) and Transformer 1 (9%).

In addition to the issue of repeated descriptions being generated for LSTM and Transformer 1, we also found that the generated captions for both models contained repeated words for some samples, such as shown in **Figure 10**. In view of these limitations, we further examine our results under the beam search approach described in Section 5.3.2, to determine whether improvements in performance can be obtained. The results for beam search are reported in **Table 5**. We observe that while the use of beam search has resulted in improved performance for LSTM and Transformer 1, it did not result in substantial improvements for Transformer 2. This is likely because Transformer 2 already generates a large proportion of unique captions under greedy search and does not face the issue of repeating words within generated captions.

MODELS	LSTM	TF 1	TF 2
SEARCH ALGO	Greedy	Greedy	Greedy
ACCURACY	0.685	0.225	0.746
BLEU-1	0.147	0.201	0.069
BLEU-2	0.126	0.151	0.027
BLEU-3	0.136	0.153	0.018
BLEU-4	0.112	0.124	0.013
ROUGE-L F1	0.194	0.256	0.148
ROUGE-L PRECISION	0.324	0.381	0.278
ROUGE-L RECALL	0.144	0.201	0.109
% UNIQUE DESCRIPTIONS	2%	9%	63%

Table 4: Model Results Under Greedy Search

MODELS	LSTM	TF 1	TF 2
SEARCH ALGO	Beam	Beam	Beam
ACCURACY	-	-	-
BLEU-1	0.161	0.241	0.072
BLEU-2	0.133	0.180	0.030
BLEU-3	0.140	0.184	0.021
BLEU-4	0.114	0.150	0.012
ROUGE-L F1	0.210	0.262	0.142
ROUGE-L PRECISION	0.304	0.310	0.272
ROUGE-L RECALL	0.173	0.242	0.104
% UNIQUE DESCRIPTIONS	2%	29%	54%

Table 5: Model Results Under Beam Search

Overall, from our model evaluation, we find that the different models have their respective pros and cons. While LSTM has scored relatively well on BLEU and ROUGE, it suffers from the issue of generating a very large number of repeated descriptions for different product images. An example of this is illustrated in **Table 6** (Appendix), where we see that LSTM has generated the exact same description for 3 tops in the validation set. Transformer 1 is able to capture product features relatively well, such as crew neck and long sleeves in the example, but also suffers from repeated descriptions, though less severe than LSTM.

On the other hand, the descriptions generated by Transformer 2 tend to be more dynamic while capturing key product features. As seen in **Table 6** (Appendix), Transformer 2 has generated 3 different descriptions for the 3 tops. This dynamic behaviour could explain its smaller BLEU and ROUGE scores as compared to LSTM and Transformer 1 when comparing the generated and actual descriptions. However, as a result, Transformer 2 might occasionally incorrectly identify the categories or features of several clothing items. For instance, in **Table 6** (Appendix), we observe that Transformer 2 has incorrectly described the third top as a jacket.

Given that the objective of our project is to develop an automated clothing product description generator that can be implemented on e-commerce platforms, it is important that the generated descriptions are varied in order for the generator to be useful and value-add to businesses. As such, balancing between our reported performance measures and human judgement, we conclude that Transformer 2's performance would be preferred over LSTM and Transformer 1. However, improvements are required to further enhance performance which will be discussed in Section 7 and 8.

In the following sections, we will use Transformer 2 with greedy search as our chosen model for further analysis on model performance and explainability.

6.2 Performance Analysis

To better understand our model's performance across brands, we have plotted the mean ROUGE-L F1 score of the various brands by gender, against their representation in the training set, shown in **Figure 11**. We observe that performance tends to be better for brands that have a larger number of samples in the training set (i.e. H&M, Forever 21, C&A). As description styles can vary widely across brands, our model might be predominantly learning from the main brands in our dataset, and thus performing less well on the brands with smaller product ranges.

6.3 Case Studies

In order to better understand our model's behaviour and identify whether the captions are being generated from the right features extracted from the images, we plotted

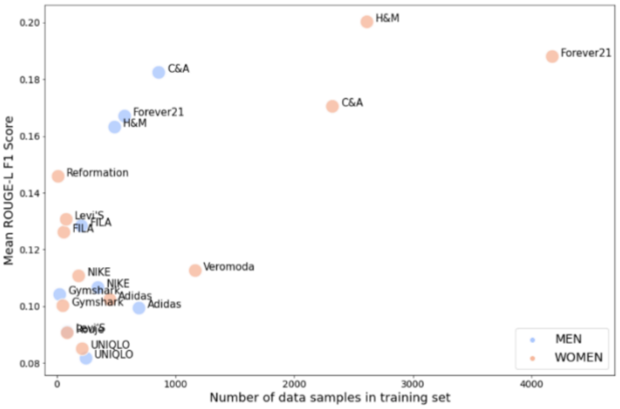
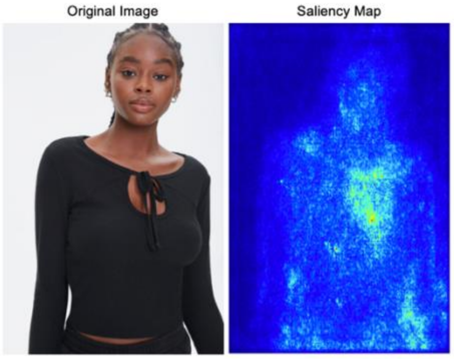


Figure 11: Scatterplot of mean ROUGE-L F1 score of brands against number of data samples in training set

saliency maps for several examples from the validation set. Saliency map is a gradient-based method of identifying relevant features from the image, whereby a large gradient in the area would indicate higher relevance in the caption generation task.

An example is shown in **Figure 12**, where our model is able to identify a tie design on the neckline. We observed from the saliency map that important features are concentrated on the correct portion of the image. This indicates that our model is able to learn and generate the caption considerably well.

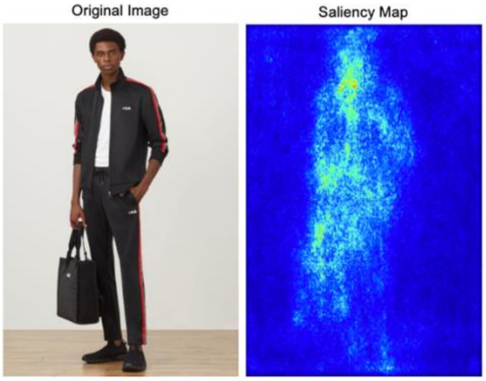


Actual Caption:
"a ribbed knit top feature a self tie round neckline chest cutout and long sleeves."

Generated Caption:
"top with a detachable tie belt and long sleeves"

Figure 12: Example of Generated Caption and Saliency Map

On the other hand, we also observed that our model may focus on different parts of the image as compared to the actual caption. One example is shown in **Figure 13**, where the generated caption described the jacket instead of the pants. This is further confirmed by the saliency map where a significant portion of important features are concentrated on the upper-body portion of the image. This is an example of the case where the generated caption is considered incorrect when compared to the actual caption, however, as we have observed, this is in fact a correct caption as it managed to describe the jacket quite well. A possible refinement could be applied to the input image data, where relevant parts of the image are cropped in order to remove redundant features which would distort our model's learning ability.



Actual Caption:
"a laid back yet luxurious look is achieved with these track pants that call back to our archives while also staying relevant to present style"

Generated Caption:
"the it sportswear jacket are made of soft woven fabric and a timeless design"

Figure 13: Example of Generated Caption and Saliency Map

7. Limitations

We recognize several limitations present in our project which we intend to further explore and improve on. Firstly, due to the complexity of our models and the size of our dataset, experimentation on different model architectures and hyperparameters are considerably time consuming. Although the models and their corresponding hyperparameters presented in this report are optimised to some extent, given more time, we could conduct more experiments and extend more training epochs to ensure our models are well-converged.

Furthermore, as described in Section 6.2, our model is predominantly learning from brands with larger representation in our dataset. Although it is ideal to acquire a balanced dataset from all brands, it is challenging in reality as not all brands have large inventories and quality captions for all products. As we have observed in the data collection phase of this project, many brands use generic and uninformative captions on their online stores. In addition to data size, another limitation for acquiring dataset from different brands is that caption styles across brands are widely varied therefore adding additional noise to the model.

Lastly, as shown in **Figure 13** previously, full-body shots are common for fashion product images, especially when brands are showcasing other matching products. Therefore, generated captions might be based on the wrong portion of the image which makes it difficult to evaluate our models without conducting manual checks.

8. Conclusion & Recommendations

In view of the current limitations of our project, we have several recommendations for a future study. Firstly, given that caption styles can vary widely across brands, clothing and gender categories (e.g., dresses vs shorts, sportswear vs casual wear), we could consider developing models separately by broad categories. This will reduce the variability of data within each category, such as the learnt image features and vocabulary size, and could thus result in improved model performance where descriptions generated for products are more relevant to their clothing category. In order to develop category-specific models, more data will have to be collected for the specific style or category of interest. Data augmentation for text can also be considered to further increase data size and enhance generalizability, such as synonym replacement and/or reshuffling of sentences.

Secondly, product images could be further processed by cropping only the product being sold (e.g., top or bottoms) from full-body images. The background of images can also be removed. These additional steps will ensure that the models are trained on the correct portion of the images for the product in question.

Lastly, we could also consider other state-of-the-art transformer models, such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer 3 (GPT-3), to examine whether further improvements in performance can be achieved with these models.

While natural language generation is still a developing domain, we recognize that it is challenging to achieve full automation to generate fluent product descriptions. As demonstrated in our project, we have achieved reasonable performance accuracy and quality in describing clothing product images. Our model outputs could therefore serve as product description suggestions for e-commerce fashion retailers to curate from and spark creativity, to ensure that the descriptions are suitable and in line with their brand image while speeding up the editing process at a fraction of the cost.

References

Holmes, A. (2019, Dec 16). The 2010s were the Decade Amazon took over the world. Retrieved from https://www.businessinsider.com/amazon-decade-review-2010s-growth-2019-12

Zeng, M. (2018, Sep). Alibaba and the Future of Business. Retrieved from https://hbr.org/2018/09/alibaba-and-the-future-of-business

Charm, T., Grimmelt, A., et al. (2020, Oct 26). Consumer sentiment and behavior continue to reflect the uncertainty of the COVID-19 Crisis. Retrieved from https://www.mckinsey.com/business-functions/marketing-and-sales/our-insights/a-global-view-of-how-consumer-behavior-is-changing-amid-covid-19

Octoparse. Accessible at https://www.octoparse.com/

Selenium. Accessible at https://www.selenium.dev

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

ImageNet Database. Accessible at http://image-net.org/

Pennington, J., Socher, R., Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Alammar, J. (2018, Jun 27). The illustrated transformer. Retrieved from http://jalammar.github.io/illustrated-transformer/

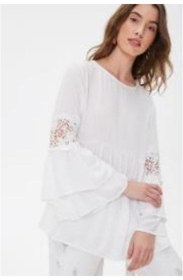
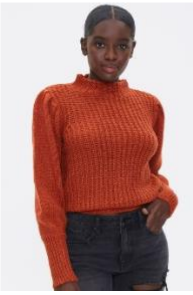
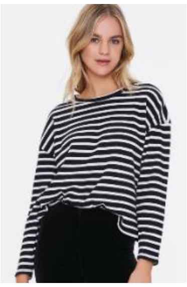
Gautam, T. (2021, Jan 20). Implementation of Attention Mechanism for Caption Generation on Transformers using TensorFlow. Retrieved from https://www.analyticsvidhya.com/blog/2021/01/implementation-of-attention-mechanism-for-caption-generation-on-transformers-using-tensorflow/

Zhang, A., Lipton, Z. C., Li, M., Smola, A. J. (2020). Dive into Deep Learning. (Section 9.8 Beam Search). Retrieved from https://d2l.ai/chapter_recurrent-modern/beam-search.html

Appendix

Github link to Python codes and dataset: https://github.com/Gami-Hiro/BT5153-GroupProject-Team3

Table 6: Illustration of the difference in descriptions generated by the 3 models with 3 clothing examples

Product Image	Model	Description
Top 1 	LSTM	a knit top featuring a v neckline long sleeves and a crew neck
	TF 1	a knit tee featuring a crew neck long sleeves and a crew neck
	TF 2	neck top in soft cotton jersey with a round neckline and long sleeves
	Actual	a woven top featuring a round neckline tiered long bell sleeves with a floral crochet panel shirred accents on the waist seam keyhole back with button loop closure and a relaxed fit
Top 2 	LSTM	a knit top featuring a v neckline long sleeves and a crew neck
	TF 1	a knit top featuring a crew neck long sleeves and a crew neck
	TF 2	neck pockets and hem give this jumper a feminine look
	Actual	a knit sweater featuring a ribbed bodice and sleeve cuffs long sleeves shirred shoulders and a mock neck
Top 3 	LSTM	a knit top featuring a v neckline long sleeves and a crew neck
	TF 1	this long sleeve top with a relaxed fit and a classic fit
	TF 2	jacket in woven fabric with a round neckline and long sleeves
	Actual	a knit top featuring an allover striped pattern dropped long sleeves and a round neckline

Note: Actual descriptions shown in the table are after data cleaning steps described in Section 4.3