Caption Generation for Amazon Products Using Weak-Supervision

Ian Raphael Sim Ek Yan Ma Yuan Nicole Lee Zhi Ying Yong Zhun Hung

National University of Singapore

Abstract

Given the boom of online e-commerce in the recent years, it is important for individual merchants to stand out by conveying accurate information about each product or item that they sell online. This is particularly important in e-commerce where the customer is unable to physically assess such properties of the product. It is therefore vital for these online stores to have accurate and compelling descriptions of their products which customers can use to make an informed decision before committing to a purchase.

We see a potential in automating the process of copywriting using data inherent in the product. These include pictures, category data, metadata and short titles of the product. These features are then parsed into a machine learning model trained to generate and output a cohesive and compelling natural language paragraph describing the product with great and accurate detail.

Dataset

Our dataset is adapted from the Amazon product/review data dataset, published by McAuley and updated by Ni (UCSD) [1]. The dataset consists of 233.1 million Amazon product reviews spanning May 1996 - Oct 2018 and includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs).

Since our primary focus is on using images and textual data to generate descriptions, we remove the unnecessary fields such as reviews and brand and retain only those which are useful to our model. These fields are listed below:

- Product Images (URLs of both High and Low Resolution Image of each Product)
- Categories (List of categories the product belongs to)
- Description (Short text description of the product)

Additionally, the raw data is over 20GB in size, which far exceeds what we are able to effectively train given the short time frame and limited GPU resources. Therefore, for this project, we uniformly sampled data from each of the 25 product categories, resulting in a final dataset with just over 7000 entries.

Data Preprocessing

- **Image Data** We perform image normalization by first resizing all product images into uniform dimensions of 320x320 pixels. We chose 320 pixels as ResNet reduces the image by a factor of 32 in the output. We then normalize the values of each pixel in the image to prevent overflow during the convolutions or backpropagation. Image augmentation and transformation is also peformed on each product image to increase the variation of training data. This is done using random flipping, scaling and skewing.
- **Text Descriptions** We initially trained the model over the complete description data which consists of multiple sentences in a paragraph. However, we noticed that this made it much harder for the model to interpret the complete meaning, as the paragraph was too long and the complexity was too high. Therefore, we decided to split the paragraphs into shorter sentences which can be better understood by the model. Sentence locations are also fed into the attention block with the intuition being that different area of the image data should correspond to different sentences in the description paragraph.
- **Categorical Data** Each product is currently modelled as a one-hot vector where each entry in the vector corresponds to 1-of-25 categories.

Methodology

The original encoder-decoder model from the show, attend, tell paper by Xu et. al. forms the foundation of our model. However, in order to combine the various forms of image, text and categorical data, we augment the model with two additional components. These are the textual description data and the one-hot categorical data. Data from the description, which consists of short sentences, was first cleaned, pre-processed and tokenized using an embedding layer. Subsequently, it is combined with the outputs of the encoder model and categorical input which is then fed into the LSTM decoder model. In order to preserve the positional data for the attention mechanism, sentence indices are also stored.

We propose and experimented with three different architectures, as labelled in Figure 1 below.

1. Solution 1 simply concatenates the output of the description embeddings with the attention block and combine it with the one-hot categorical data before feeding it into the decoder
2. Solution 2 introduces connections from the categorical and description outputs into the attention block inputs, thereby allowing the model to attend to different parts of the text or category data.
3. Solution 3 utilizes zero-shot learning to try to predict previously unseen or missing data.

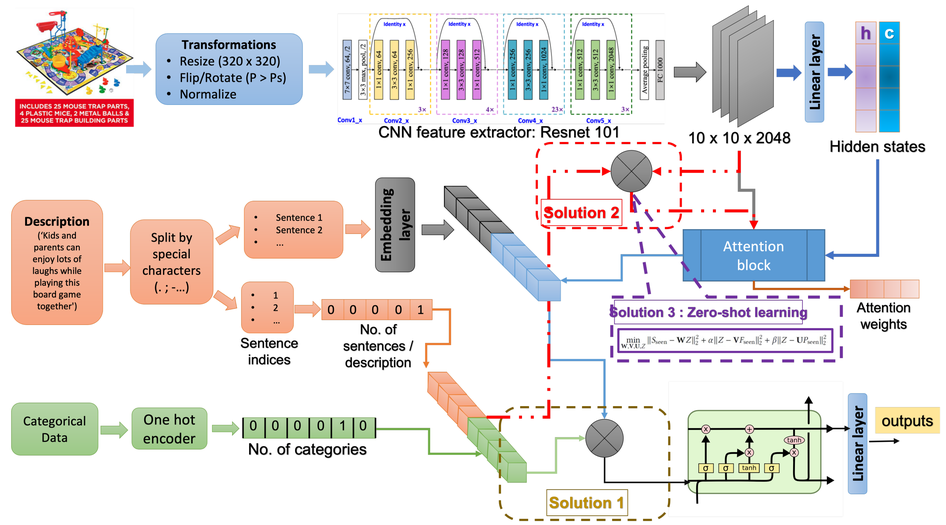

Figure 1. Architecture Diagram with Proposed Solutions

Encoder

- Uses a pre-trained ImageNet ResNet-101 model for the encoder
- Removed linear and pooling layers of the ResNet-101 model
- Incorporated an Adaptive Average Pooling to allow for variable image sizes. This is quintessential in the business context where users often do not upload images of the same sizes.

Attention

- The attention block uses inputs from the image encoder and LSTM text decoder to attend to regions in the image corresponding to the next predicted text token at each LSTM cell.
- In solution 2, the attention block additionally has inputs from the categorical vector which are combined with the image data with the intuition being that the attention model can attend over different product categories differently.

Decoder

- Uses a standard LSTM network to output a series of words
- Initial Input to the decoder is
  - Solution 1: Output of Attention Model
  - Solution 2: Output of Attention model and Embedded Category/Description data
- Output of each LSTM cell is a softmax normalized probability distribution over the complete vocabulary and the word with the highest probability is chosen for that cell

Motivation

To evaluate if combining multiple data mediums (image, text, categorical) can improve performance in image captioning on weakly labelled data when applied to product copywriting

Experimental Results and Discussion

The BLEU (bilingual evaluation understudy) score is an industry standard benchmark used for evaluating the performance of natural language models. BLEU is calculated by comparing the model output to a reference human-generated sentence and scored according to how closely it resembles the reference text. A score from 0-0.2 indicates poor performance, 0.2-0.5 indicates proficiency and above 0.5 is generally considered high quality.

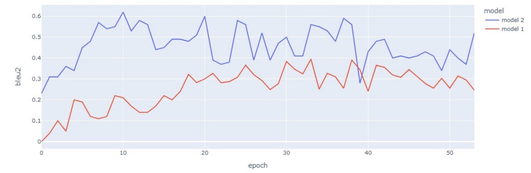

Figure 2. BLEU-2 Score of Experiments

The BLEU-2 score for **Solution 1** varies between 0.2 to 0.4 reflecting decent, but not great quality of natural language generation. Results for **Solution 2** are more promising, with a higher average BLEU-2 score ranging from 0.3 to 0.6. This represents a significant improvement and could be considered high quality on the upper end. However, there exists a high variance among different products with some products generating higher quality description than others.

Furthermore, in some cases, we observed there were missing categorical or image data. Missing data is a common challenge faced by many machine learning problems. In order to address the issue of missing data, we introduced zero-shot learning as one of the possible extensions in Solution 3. Instead of trivially concatenating the category embeddings and image embeddings as shown in Solution 2, Zero-Shot learning introduces a latent variable Z as the "bridge" to fuse the semantics of both image and category embeddings. This latent Z computation can be solved as an optimization problem, with minimizing $\|L2\|$ between latent variable Z and learnable matrices W, U, V and each corresponding embedding as the objective.

Next Steps

Given the experimental results, the team will be looking into making the following improvements to the model:

- **Zero-Shot Learning**: Train the model to generate feature representations of unseen products and thereby further improve generated captions
- **Beam Search**: Currently, the next word in the decoder sequence is chosen greedily, based only on the local probability of that cell. We could potentially introduce beam search to optimize the selection of the next sequence of words and maximize over the joint probability. This could potentially give better results.
- **Larger Dataset**: We are limited in scope and scale by the lack of access to high-power computation resources as well as time constraints. As we are currently using only 0.003% of the available data, it stands to reason that performance will be much improved with the larger complete dataset.

References

[1] Jianmo Ni, Jiacheng Li, and Julian McAuley. *Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects*. 2019.

[2] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[3] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2015.

[4] Xueying Zhang, Yanyan Zou, Hainan Zhang, Jing Zhou, Shiliang Diao, Jiajia Chen, Zhuoye Ding, Zhen He, Xueqi He, Yun Xiao, Bo Long, Han Yu, and Lingfei Wu. Automatic product copywriting for e-commerce, 2021.