**Intelligent Marketing Text Generation**

Multimodal text summarization applied in the fashion industry

Authors:

**Jingxuan Li**

**Tu Xue**

**Wenxuan Tang**

**Vivian Chan**

**Fenny Jin**

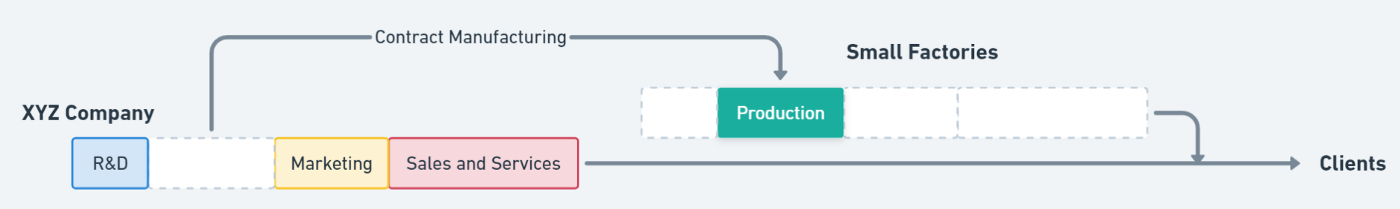
**Guanyuan Shuai**

**Audiences: XYZ Company, a player in the fast fashion industry with a focus on online sales channels**

# **1.Introduction**

## **1.1. Business background**

Our client, XYZ Company, is an apparel retail company that focuses on online channels. The company is one such international fast fashion brand that extends the best fashion wear including women's wear, men's wear, children's clothes and apparel, trendy accessories, and other fashionable products and accessories. XYZ serves as an intermediary linking a large number of small factories with production capacity to consumers with a passion for fashion. It receives a certain number of orders cumulatively from the consumer side and then forwards them to the factory for centralized production, and then sends the finished products to individual consumers.

XYZ Company’s business chain structure

## **1.2. Project background**

In order to enhance the company's competitiveness in the marketplace, company XYZ approached us to develop a intelligent service that can enrich the product descriptions that were already available on the website: converting keyword or tag style descriptions and pictures of products into more fluid and smooth marketing copy as shown below :



With approximately 40,000 to 50,000 new products updated on the website every week, there was a big need to automate part of the copywriting process. The purpose of this project is to help the company build a model for intelligent marketing copy generation. We will continue to introduce the specific product concept and features later.

# **2. Discovery summary**

For XYZ's specific needs for the product we analyzed and summarized them by conducting three Need-Finding activities, which are naturalistic observation, participant observation and interview.

After conducting a **naturalistic observation**, we found that using more fluid marketing text to describe items has become a trend, especially on the Taobao platform in the Chinese marketplace. And even on overseas online shopping platforms such as Amazon, more and more merchants are providing more detailed and interesting descriptions on their product pages. These descriptions actually contain a large number of keywords about the product.

Then we also organized **participant observation** and found that as consumers there is also a growing tendency to buy clothes online. One of the reasons for supporting consumers to buy online is the ease of finding the required and detailed product information. The detailed product information included multiple perspectives to de-categorize the clothes. However, manually sorting products by dozens of categories is time-consuming and has a high error rate. At the same time, we also find that products with nice descriptions attract our attention and increase our chances of buying. Sometimes the description is not even very detailed, but if it is presented in the form of a sentence, it will also increase our positive feeling about the item.

We have conducted demand communication and in-depth research with XYZ. From the **interview**, we found XYZ Company pays a lot of attention to how the details of the products in the order are communicated to the manufacturing plant, so for each item a lot of labeled text is generated during the design and production process. At the same time, a large number of new models are added to XZY's website every month. In the course of its fast growth, XYZ found that writing product descriptions for each item could be very labor-intensive.

In the above analysis, we summarized the current detailed requirements of XYZ company for this project. First, our model should output the corresponding readable description of products. Then, the data sources for description generation are product images and product labels. In addition, the generated text should include most of the product's features and selling points. Finally, the model needs to have a speed that allows it to generate marketing text for a large number of new products in a short period of time.

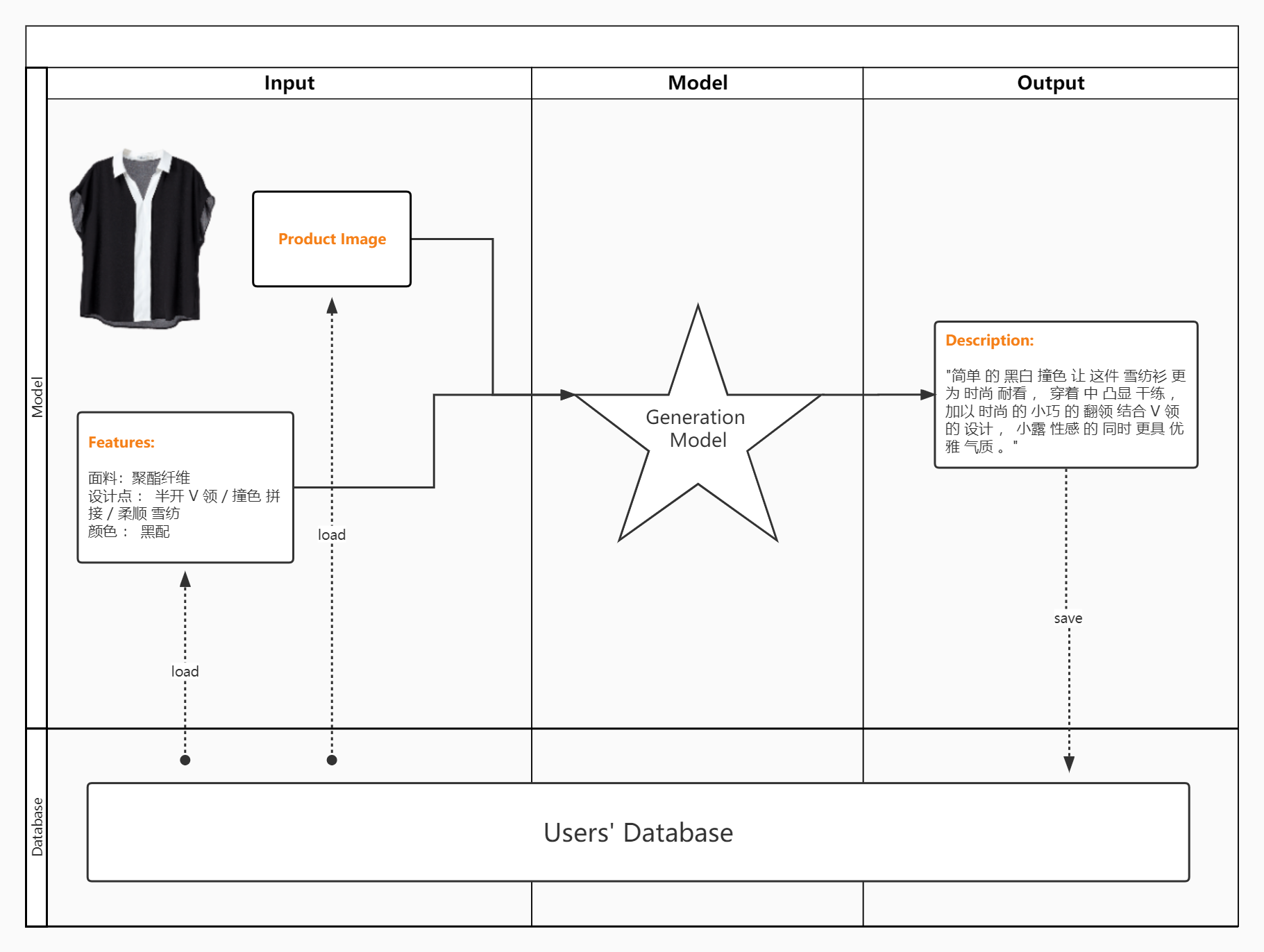
# **3. Brainstorming summary**

Using our discovery as a starting point, we brainstormed an idea for an interactive online platform that would display marketing text generated based on the clothing images and product features selected by a customer in real time. The text-generated model is made by combining computer vision and natural language processing deep learning models and enriched by our trending keywords web scraping machine learning model.

After the customer imports the images of clothes, the computer vision model will extract important features from the images and add these to the NLP model to make it more robust. The NLP model uses both the features from the computer vision model as well as the text information from keywords to generate clothing descriptions.

As fashion trends change over time, it is essential to enable our model to learn more current trendy keywords that are likely to boost sales and SEO rankings, such as 'chic', 'chromatica', and 'camp', so we will also retain another machine learning model that will scrape marketing information and keywords from social media such as Instagram, Twitter, and other online shopping platforms so that over time, our model will learn more trendy words, and add it to our text summarization NLP models to boost overall performance.

# **4. Prototyping**



We are trying to design a solution to generate the clothes description automatically. The customers can select and import the clothes images and its features or labels, then the model will generate the description of the clothes for marketing purposes.

Our solution is connected to an existing database to facilitate the reading and storage of the model's data. Customers can import a large number of image files with any format at one time and the output will be a file that contains the description for each clothes. This product will save the preparation time for customers to fasten the promotion of their own products.

# **5. Evaluation: user interviews on the design**

We also presented the prototype to the end-user of XYZ company and they said the whole process was very clear and mentioned that it would be a very helpful tool not only for company XYZ but also for many newly established fashion companies. Respondents also mentioned the importance of tag accuracy in this model, as often users go through the text with a target to find the corresponding keywords. Therefore, it would be useful to provide accurate labels or images of products from different angles to help the model generation. Interviewees felt that the marketing impact of the text could also be improved if some popular words could be included in the description. This trending vocabulary could be updated in real time to adapt to changes in the market. Finally, the user also mentioned that if an option to control the length of the generated text could be added, it would be more adaptable to different presentation scenarios, but it is not necessary.

# **6. Final product solution design**

## **6.1. Final production solution**

After designing a prototype and getting feedback from users, we settled on a final solution. Our intention was to build a product marketing text generation model with four components on the input side.

* The first is the **product image**. We intend to try to input only a single image first, and if we get suitable supplementary data later and the accuracy of the model is difficult to improve with the input of a single image, then we can add product images taken from different angles.
* The second input is **product specific features and labels**, such as material, style, and pattern. The reason these features need to be manually entered in the first place is that if we want to rely solely on machine learning methods to automate labeling we may need several targeted and different mini-models, as they are difficult to identify by a single integrated model. This increases the engineering effort of the whole project. Then, because of the laws and regulations, clothing and other goods must be labeled with their material and approximate style, so it is not so difficult to obtain these features, and thus generating the corresponding text becomes an uncomplicated but effective task. Finally, there are many keywords that are the main reason to attract customers to buy products, so this attribute can be controlled manually to improve the ability of marketing copywriting to respond to market changes and market trends.
* The third input is alternative and is the **words and rhetoric that are currently in vogue**. Marketing language changes rapidly in the online environment, and different people have their own preferred text styles. Therefore, adjusting the style of marketing in a timely manner can also serve to better attract customers of different positioning.
* The last input is the **length of the text to be generated**. Depending on the application scenario, store operators can choose whether they need to generate more concise or more detailed text.

On the output side, our model can produce text close to that written by a professional marketing copywriter, which includes the main features of the product and a style that blends with popular trends of the day to better attract customers.

In addition to the modeling aspect, our products will also provide ports to connect with databases in the future to facilitate the management of big data in the context of cloud computing.

## **6.2. Business Analysis**

We found another brand in the market, Jasper, also doing smart text marketing generation solutions, so we will do a comparison with them to show the innovation of this project.

First of all Jasper's main user group is producers of online media content, who mainly produce text output for marketing purposes on social media and blogs. It usually contains a lot of philosophical stories or inspirational words to move readers, so it can be considered user-oriented. Our main customers are businesses that want to sell physical products, and our model generates text for a single product, so **our solution is more product-oriented**.

Then again Jasper's model input is only text keywords. Our big innovation on this point is to generate text through images, which can mobilize both verbal and image parts of the reader's brain and make the product more memorable.

One of the points that we need to improve, inspired by Jasper, is the data acquisition part; Jasper's data sources are massive web resources, because simple text resources are easier to find than text-image pairs. So our subsequent work needs to be more focused on connecting with businesses that already have a rich database of marketing texts.

# **7. Technical Solution Design**

Based on the project proposal described in the above sections. Our major objectives are as follows.

* We design a summarizer that can automatically generate an aspect-aware textual summary for clothing products by integrating textual and visual product information.
* We propose a multimodal pointer-generator network and explore various approaches to using product images in the product summarization task.
* We adopt aspect training, aspect coverage, and aspect coherence strategies, which aim to improve importance, non-redundancy, and readability, respectively.

## **7.1. Dataset Construction**



Example of data sample

As a large-scale e-commerce platform based in China, our client company XYZ provided us with a dataset containing images of 30,000 products from the Clothing categories. There are 18 kinds of products in Clothing, such as jackets, dresses, and T-shirts. Each instance in our dataset is a (product information, product summary) pair, and the product information contains an image and product descriptions. The description part includes texts such as labels and keywords for each product, as well as a product summary generated by thousands of qualified writers from the e-commerce platform. The professional auditing groups will strictly review the summaries, and only the high-quality summaries are kept.



Keyword in samples from word cloud

## **7.2. Metrics Design**

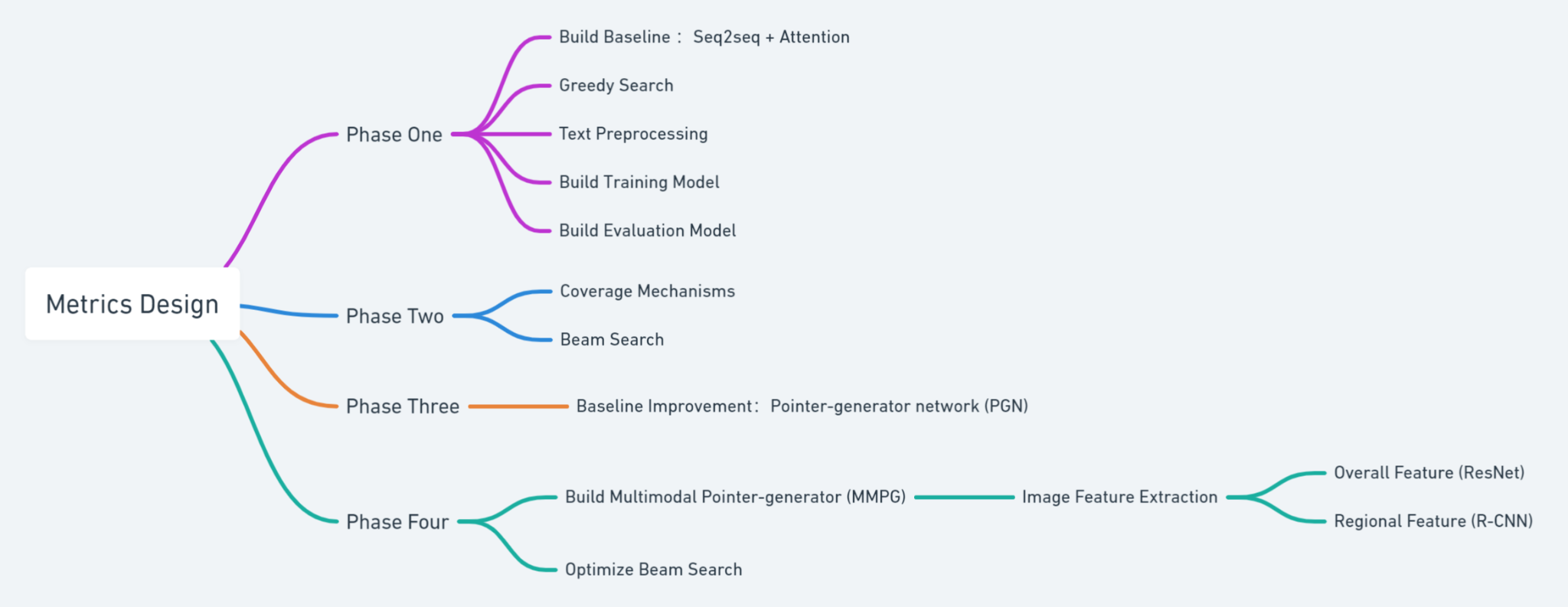
For the design of the model, we first decided that we would use deep learning frameworks. This is still a natural language processing (NLP) task, although we will eventually incorporate information from the images into the model. Based on the data situation we could use and what we ultimately wanted to accomplish, we considered this a text summarization task. Based on the data available to us and what we ultimately wanted to accomplish, we considered this to be a **Text Summarization** task. According to the output type, there are Abstractive Summarization and Extractive Summarization. Extractive Summarization consists of key sentences and keywords extracted from the source document, and the abstracts are all derived from the original text. Abstractive Summarization allows the generation of new words and phrases to form the abstract based on the original text. Because we want to generate text that includes key information from the original text, and we want to have some novel sentences, we will use a framework that combines these two approaches methods in the NLP section. To combine the information from the images, we will also add a computer vision (CV) module to build a multimodal framework

The project technique design will be conducted in four phases. In the first phase, we will build a baseline for NLP, which is mainly based on Sequence-to-Sequence Model (seq2seq) and Attention Mechanism. For text generation, the baseline model will use a greedy search. Other work in this phase will include text pre-processing, building a training and evaluation pipeline, and other steps.

For the second phase, we will add **Coverage** mechanisms to reduce the repetition of generation and also apply the **Beam Search** approach to our NLP model, which will increase the accuracy of language processing.

Phase three will be an improvement for our baseline by applying a pointer-generator network (PGN) to our NLP model. PGN will adjust the weight of each word to implement a word copying mechanism, then the final output description will contain the important words in the original text.

In the last phase, a multimodal pointer-generator (MMPG) will be created to complete our project technique design. The MMPG will combine the previous NLP model and a new CV model to improve our model inputs. We will extract both overall and regional features from each image to assist the NLP model to generate some major features.



Metrics Design WorkFlow

### **7.2.1. Product Aspects**

The product aspects are some characteristics of a product, so identifying the dominant aspects of a product is beneficial to generating an attractive summary. Before modeling product aspects, we should acquire a practical approach to recognize product aspects. Based on a preliminary observation of our dataset, we find that a complete product description or a product summary may contain information about several aspects of the product. Meanwhile, each subsentence that is divided by punctuations including commas, periods, semicolons, question marks, and exclamation marks always describes a specific aspect. For such a subsentence, we can identify its aspect using some keywords.

### **7.2.2. Aspect Coverage**

We employ an aspect coverage mechanism to eliminate aspect redundancy, which is expected to increase the recall for valuable aspects. Word-based coverage mechanism aims to avoid generating repetitive words by tracking the attention history and penalizing the repetitive attention to the same position of the source text. For our aspect coverage model, we record the attention history for all the product aspects, and then the decoder is discouraged to pay the current attention to the aspects which have received enough attention before. Specifically, we maintain an aspect attention vector , which is the sum of attention distributions over all the words belonging to aspects. Then we calculate aspect coverage vector as sum of aspect attention distributions over all previous decoder steps.

### **7.2.3. Pointer-Generator Networks (PGN)**

The pointer-generator network is a seq2seq model with a copying mechanism that can copy tokens in the source sequence into the proper positions in the target sequence. Given a source sequence x, the seq2seq model maximizes the conditional probability of a target sequence. The encoder and the decoder are the two primary components in the seq2seq model. The encoder reads a variable-length input sequence and then converts the input sequence into an encoder hidden sequence.

### **7.2.4. Multimodal Pointer-Generator (MMPG)**

The visual features based on the model’s hidden state initialization are different from a single pointer-generator network. For a general pointer-generator network, the encoder’s initial hidden states are zero vectors, and the decoder’s initial hidden state is initialized using the last hidden state for the backward LSTM. For our multimodal pointer-generator networks, given a product image, we extract the activations from the last pooling layer of ResNet-50 which is pre-trained on images as the global visual features, which we use to initialize the encoder and the decoder.

## **7.3. Metrics Evaluation**

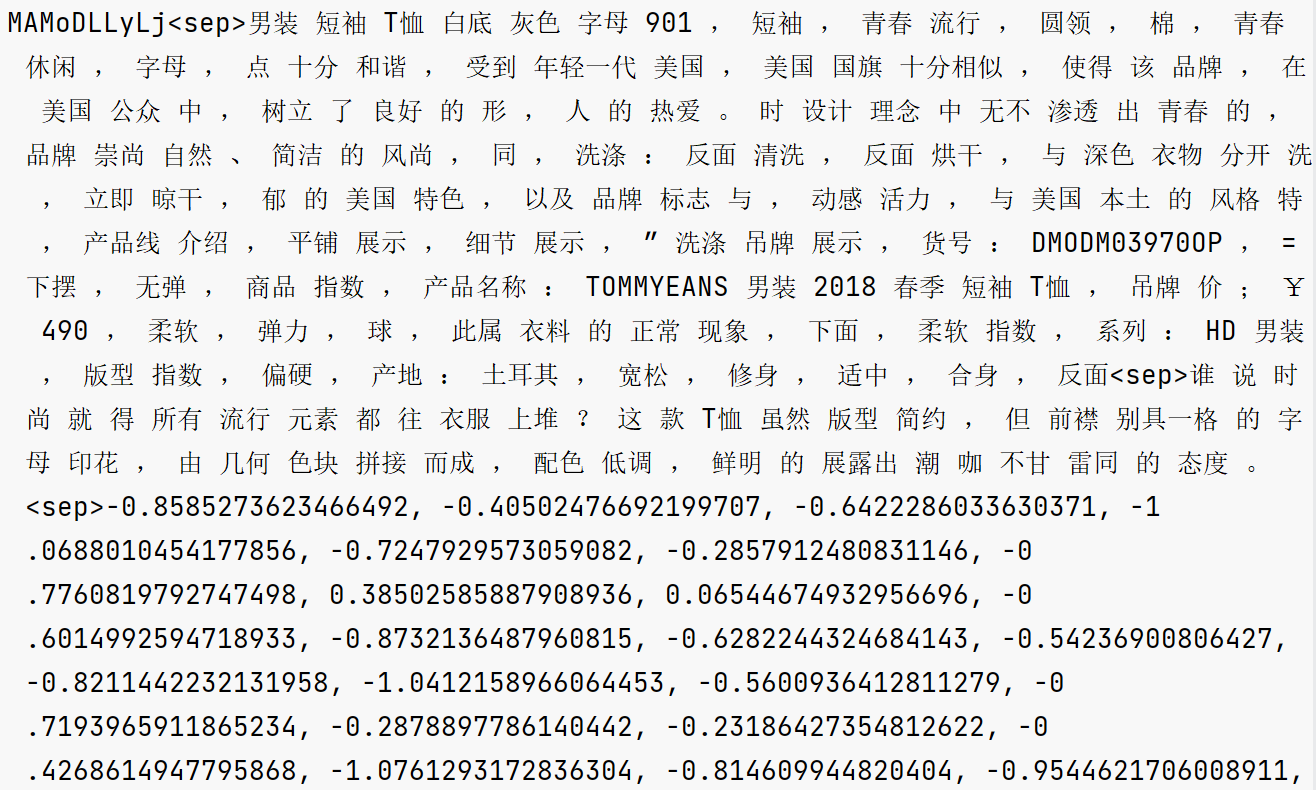
The metrics evaluation mainly consists of two parts. Based on our NLP model, we will use Rouge to measure the accuracy of output. In the first phase, the baseline model will convert the sequences following one-dimensional direction, so the results of Rouge-1, which refers to the overlap of unigram (each word) between the system and reference summaries, will represent the accuracy of our baseline model. For a modified model with coverage mechanisms and beam search in phase two, we will use Rouge-2, which refers to the overlap of bigrams between the system and reference summaries, to evaluate the accuracy of the optimized model. Besides these designed metrics, we will go through the results manually to check the fluency of output sequences.

# **8. Solution development and results**

## **8.1. Phase 1 Data processing, construction of Baseline models and basic training pipeline**

### **8.1.1. Data Processing**

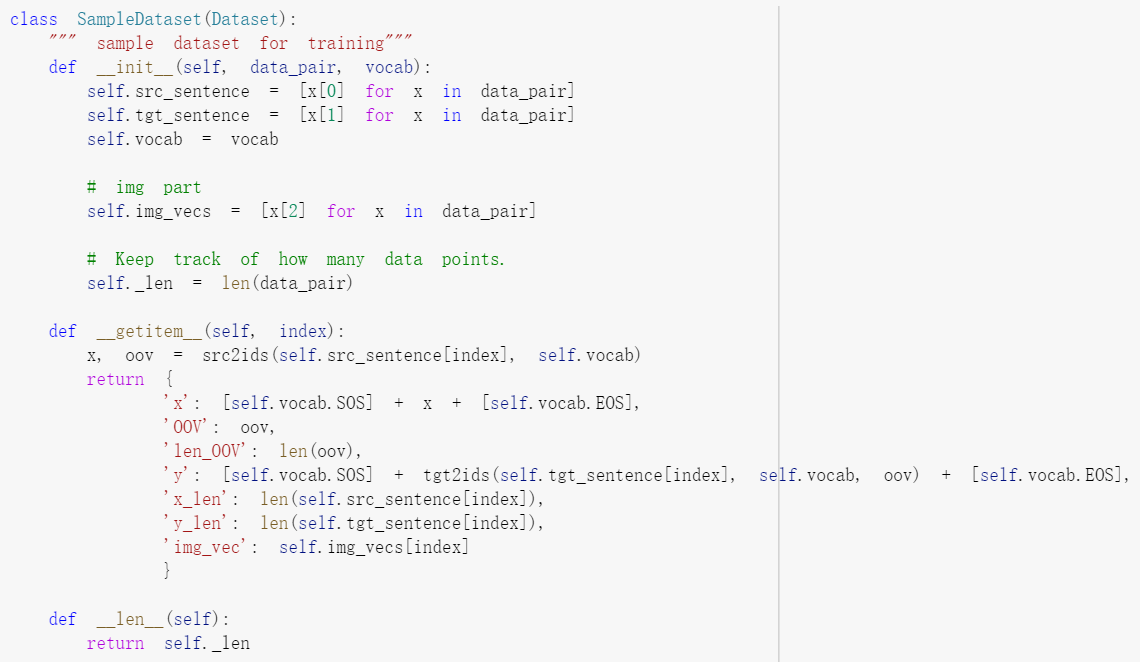
We started by concatenating the data within each sample with “ <sep>”. As shown in the figure below, each sample has 4 pieces of data, in order, the unique number with the same name as the product image file, the keywords and tags used as input to the model regarding the product description, the marketing text of the professional writers used as tags, and the vector representation of the images. For the embedding of images, we use the ResNet101 model, which automatically converts the image into a 1000-dimensional vector. We divided the entire sample set (30,000 samples) into a training set (70%), a development set (20%), and a test set (10%).



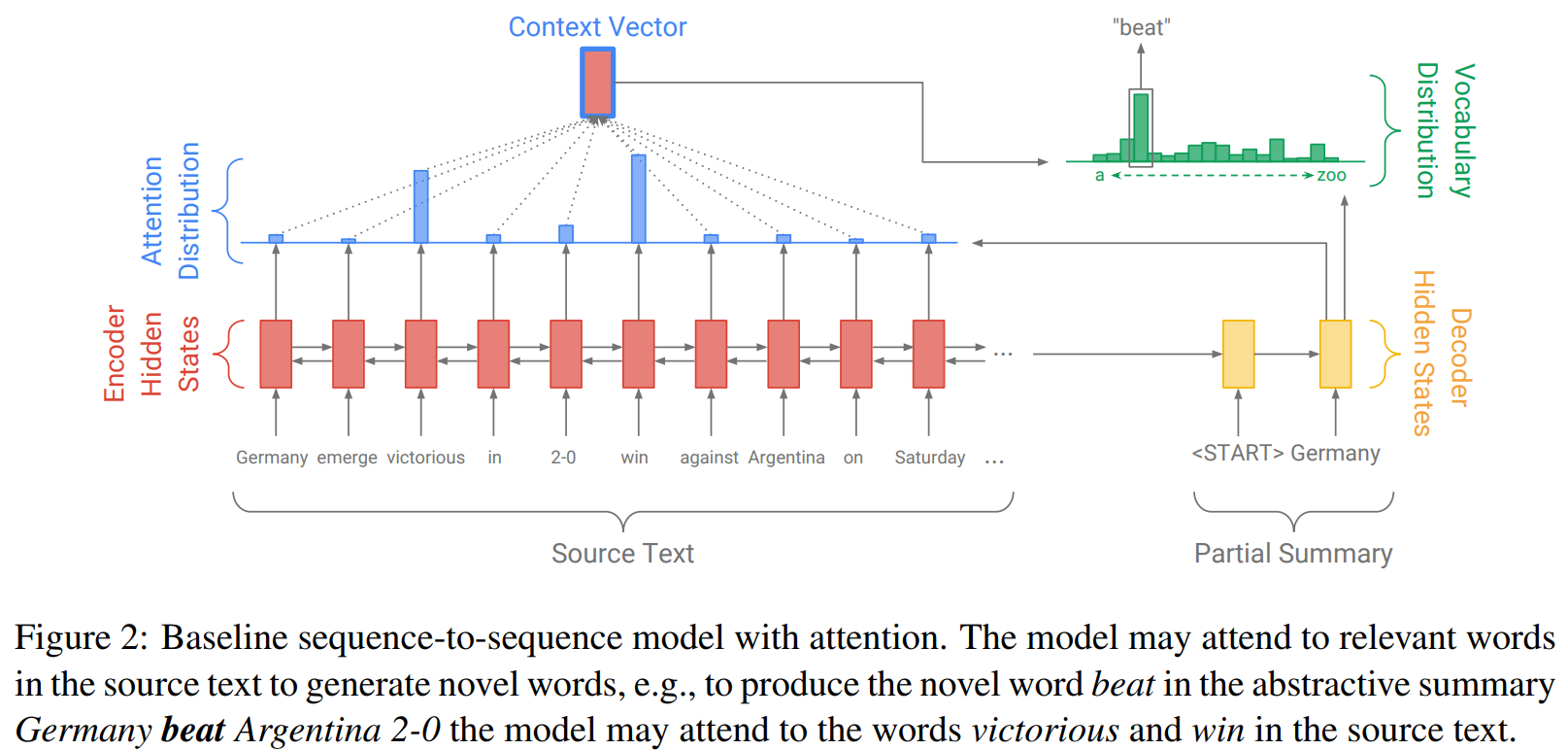
Single sample data structure

In an NLP task, we need to first tokenize our texts into each token, then turn each token into some number index that can be read by the model. We use a vocabulary dictionary to turn a token into a number index.

In the PyTorch framework, it is convenient to use “dataset” and “dataloader” to load and read data during training, and here we customize SampleDataset for this purpose as follows.

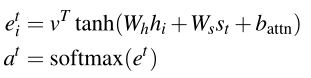


### **8.1.2. Baseline**



Our baseline model is shown in the figure below. The tokens in the article are fed one by one into the encoder (single-layer bidirectional LSTM), producing a series of encoder hidden states h\_i. At each step t, the decoder (single-layer unidirectional LSTM) receives the word embedding of the previous word (at training time, this is the previous word of the reference abstract, and at testing time, it is the previous word issued by the decoder) and has the decoder state s\_t.

Note that the distribution a\_t is calculated as follows :



The weighted summation to generate the content vector is calculated as follows :



The word distribution obtained at the decoding side is calculated as follows :



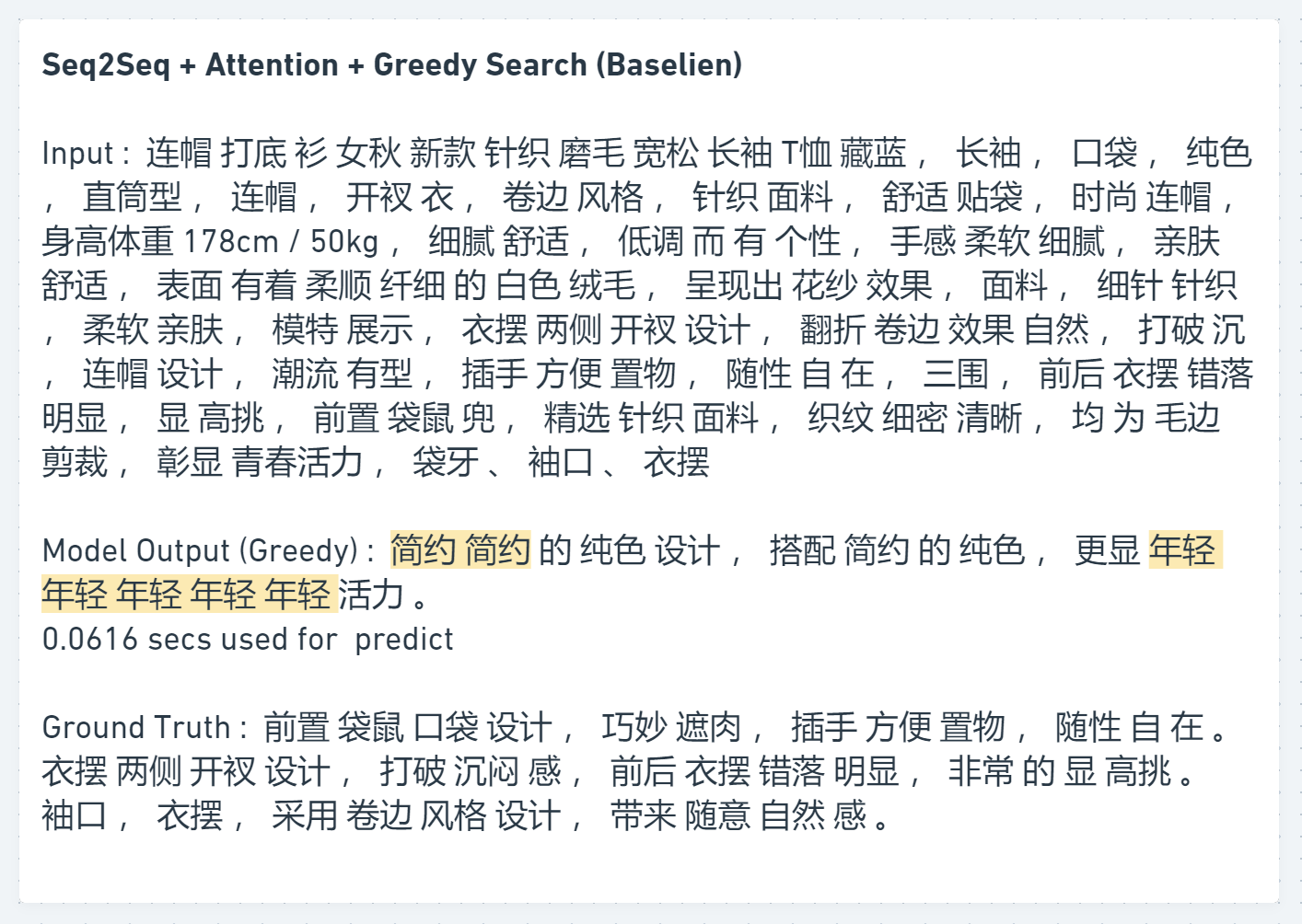
Since the text generation task is essentially a multi-classification task with sequences, for the loss function, we choose **Nagative Maximum Likehood**, which is a natural logarithmic form of the likelihood function that can be used to measure the similarity between two probability distributions.

The main body of our model consists of four parts, an encoder, a decoder, an attention mechanism, and an overall Seq2Seq model to link them together. We use the encoder to process the input sequence to obtain the output and the hidden state, and then downscale the hidden state as the initial hidden state of the decoder. For each time step, we compute attention, which is then applied to the decoder to obtain the final probability distribution.

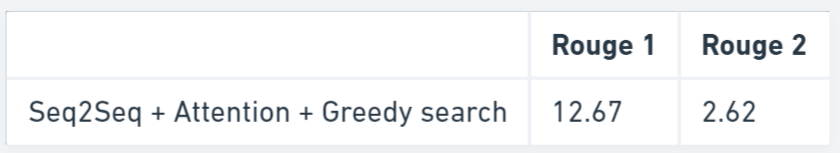
### **8.1.3. Training Pipeline**

After the main body of the model is built, we also build a training and evaluation pipeline. In each epoch of training, the models that achieve lower loss on the development set are automatically saved in a folder.

For the prediction phase, the text is generated using a greedy search algorithm, where the word with the highest generation probability is selected as the output at each moment. We can see the approximate performance of the generation with the following example.



After five to six rounds of training, the models are automatically evaluated on the test set using the ROUGE metric. The 3000 samples from the test set will go through the Predict module, and then the text will be generated with a greedy algorithm and compared with Ground Truth. The final Rouge 1 and Rouge 2 scores are output and results of the Baseline model are shown below (the higher the better).



Results are shown in percentile

## **8.2. Phase 2 Optimizing Baseline with Coverage Mechanism and Beam Search**

### **8.2.1. Coverage Mechanism**

We found a large number of duplicate words in the sentences generated by the Baseline model in the first stage, so we can refer to the Coverage mechanism to penalize the duplicate generation. So we added Coverage Vector to the model to track and control the repeat range of the source file. In our coverage model, we keep the coverage vector C\_t, which is the sum of the attention distributions of all previous decoder time steps :



Coverage vectors are used as additional input to the attention mechanism :



We define a coverage loss to penalize the allocation of excessive attention to the same location repeatedly :



### **8.2.2. Beam Search**

Because the text is generated sequentially, the output of the previous time step will be used as the input of the next time step into the model, so if the model predicts incorrectly at the beginning time step, the output as a whole will move in the wrong direction. This is why we need to use Beam Search, an algorithm that selects the top N words with the highest probability each time, and then uses these N words to predict the next time step in turn, and then selects the top N outputs with the highest probability in the output of the next time step. The top N outputs with the highest probability value among these N\*N outputs are selected to continue the prediction for the next time step. Beam Search gives the model a large fault tolerance and has been proven to be effective in lengthening the overall length of the generated sequences.

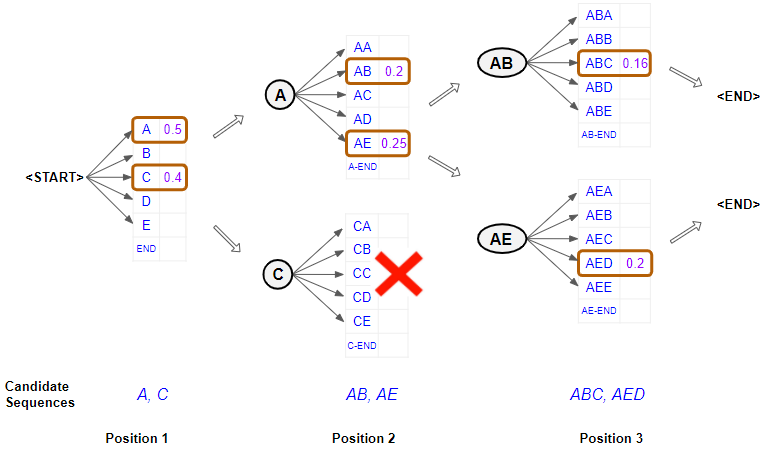
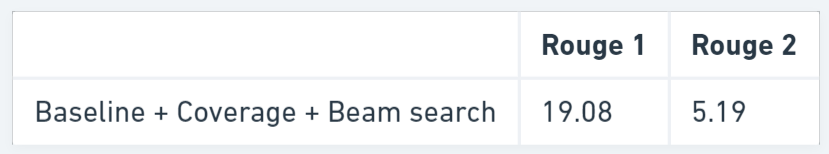


Diagram of Beam Search

After the improvements of Coverage mechanism and Beam Search, the performance of the model has been significantly improved. The repetition of words was heavily reduced. Although there are still some repetitive sentences, this is due to the size limitation of the overall word list. Since the model needs to generate N^2 results each time, the time required for model prediction is greatly increased, probably about ten times that of Greedy Search, but still within an acceptable range.

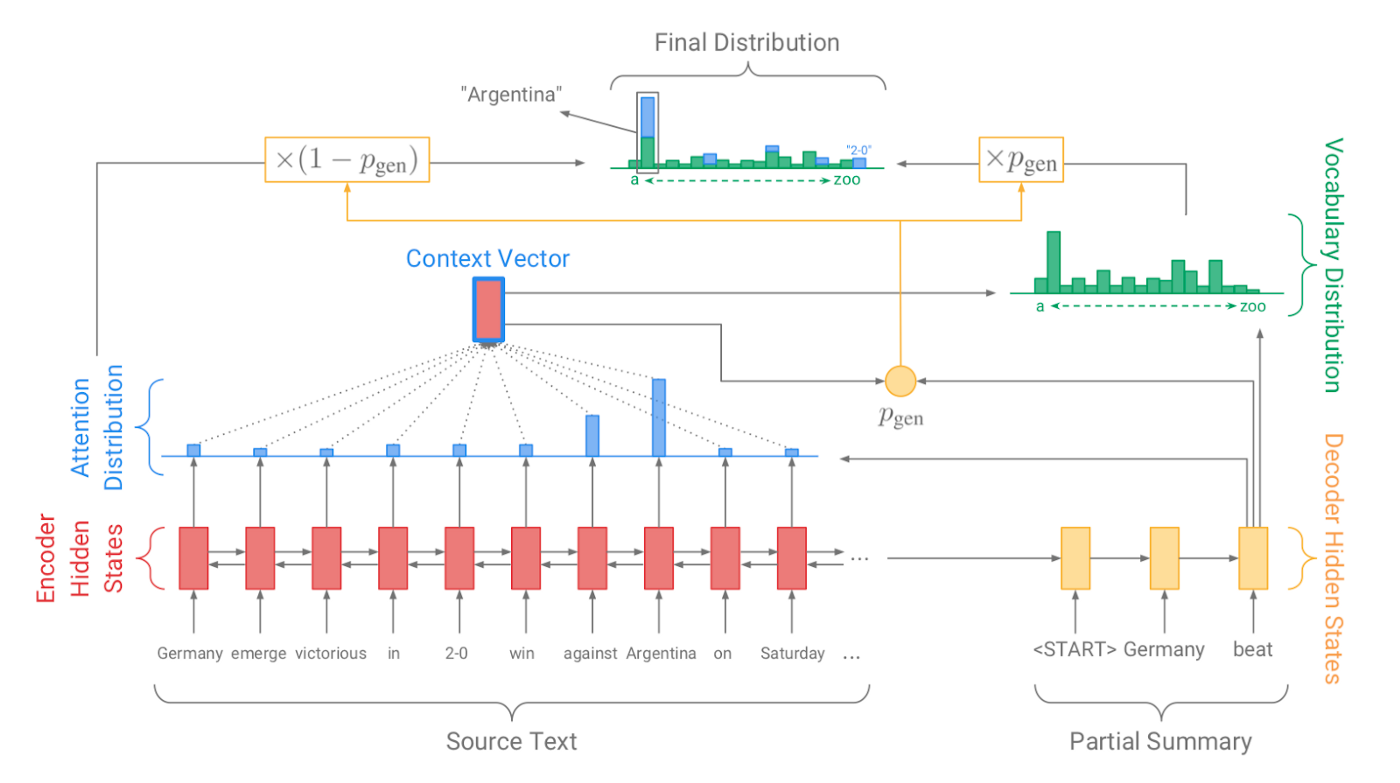


For the ROUGE metric, the model is as follows :



## **8.3. Phase 3 PGN structure**

The previous models were based on the Abstractive Summarization approach, and we also wanted to generate a vocabulary with more unique keywords about the item itself. Therefore, we introduced the Pointer Generator Network (PGN) framework in phase three. The PGN helps to copy words from the source text by means of pointers, which improves the accuracy and processing of OOV words, while retaining the ability to generate new words. This network can be seen as a balance between extraction and abstraction methods.



PGN structure

For each decoder time step, the generation probability p\_{gen} ∈ [0,1] is calculated, which weights the probability of generating words from the vocabulary rather than the probability of copying words from the source text. The lexical and attention distributions are weighted and summed to obtain the final distribution, and predictions are made accordingly. The generation probabilities are calculated as follows :



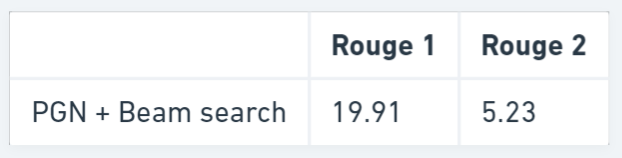
The word probabilities were calculated as follows :



The final generated copy can be shown by this example below. We can see that the brand words in the input are copied into the generated text.



For the ROUGE indicator, the model results are as follows :



## **8.4. Phase 4 Combining PGN models with image information**

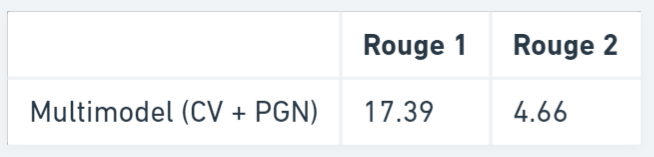
We add the image vectors to the PGN model for training as well, hoping that the information from the images can help in the text generation. We directly combine the image vector with the output of the encoder and input it to the decoder as the initial vector.



Due to the huge amount of data added with image information and limited GPU resources, we had to reduce the overall number of hidden layers of the model from 256 to 128 in order to make the training process go successfully. An example of the final generated result of the model is as follows :



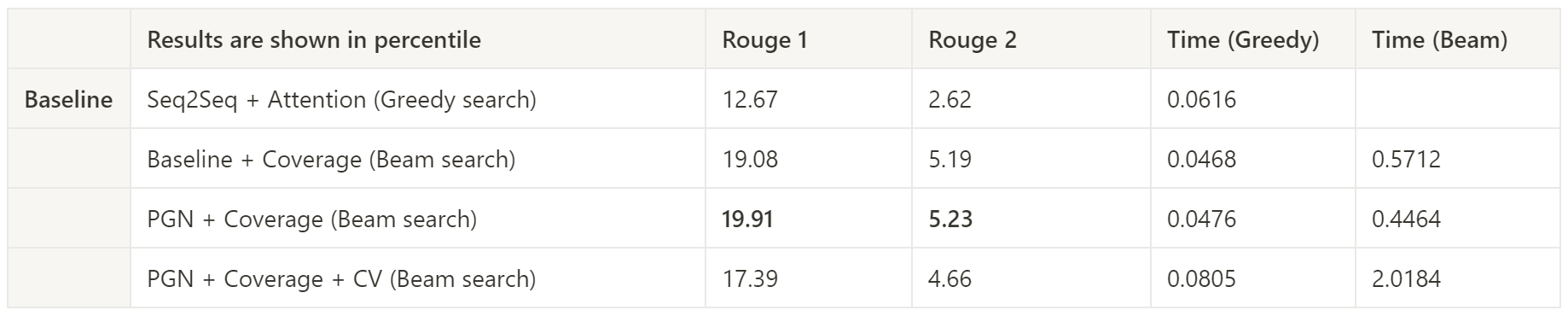
The model does not seem to capture the replication mechanism of PGN, while the phenomenon of repeated generation seems to be aggravated. We looked at the ROUGE metric as followed :



The fruitful model does not perform as well as expected, and we will try to explain why and summarize the overall results of the experiment in the next section.

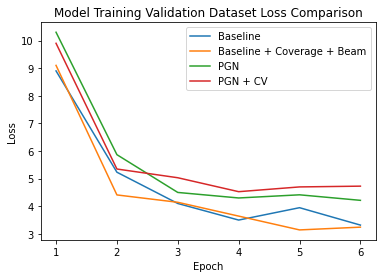
# **9. Summary**

## **9. 1 Solution Assessment**

****

We have two metrics to judge the performance of the model, one is the loss function on the validation set during training, and the other is the ROUGE metric. How should we choose?

The loss curve of the model on the validation set during the training process shows that Baseline and Baseline the modified version (Coverage+Beam) have the lowest loss, but the ROUGE metric of Baseline is not great. This is due to the inconsistency of the loss function and the evaluation metrics. Although the model can generate mathematically similar results, it may not be consistent with human reading habits. Therefore, the ROUGE metric is more representative of the model's ability by taking into account the context and the corresponding vocabulary.



The PGN model has a higher loss curve than Baseline due to the inclusion of more parameters, but the specific generation performance, assisted by the replication mechanism, results in a higher ROUGE score for the generated text.

Why does it underperform after adding the information of images? Because of the huge amount of data after adding the image information, the limitation of our computer resources forced us to reduce the number of hidden layers of the neural network during the training process, which resulted in the model not being able to fully exploit the advantages of image information, PGN and even Coverage. And because of too much information, the loss curve is difficult to reduce, and the generated text effect is not judged well by the ROUGE metric.

In terms of time spent, we can see that the time taken by the Greedy algorithm before adding the image information is basically the same level. After using Beam Search, the time required increases by a factor of 10, but it is still an acceptable range. After adding the image information, we find that the time required for both Greedy search and Beam search is much higher.

Initially we wanted to be able to input images, keywords, and sentence lengths to generate corresponding smooth and readable copy. Our model can now generate readable sentences with corresponding keywords, and the manual review works well. However, we did not build a mechanism to control the length of the generated sentences, firstly because we did not find any corresponding cases on the Internet, and secondly because we did not have enough time. Finally, our model does not perform well for image input. For this point, the first is because we don't have enough resources to train the model. The second may be because in the original technical design, we wanted to capture the local features of the images by R-CNN and assist the generation of NLP models together with the full features captured by ResNet. However, the R-CNN part has not been done yet due to time, so it may be that the local information is missing and our model does not generate well in the end.

## **9.2 Business Implication**

XYZ’s shopping experience currently supports 14 languages, including English, Arabic, French, Italian, Spanish, Thai, etc. With more than 1,000 items added to the new arrivals section daily, writing tailored descriptions in different languages has become an overwhelming burden for human labor. With the help of our text generator, it would generate thousands of marketing descriptions daily, which is equivalent to the workload of 500 editors/translators. Repetitive work is where an algorithm can step in. If patterns or keywords are already identified, it would be more economically and timely efficient to dispatch the description generation task to an algorithm. From an economic standpoint, only one person is needed to run the algorithm, and the output of the algorithm produces 10 times the results that could have been written by professionals. In this scenario, our model not only has the technical feasibility to execute this task but also cuts down the cost of human involvement. From an efficiency standpoint, our model can extract more precise and accurate information from each image and incorporate the text description with marketing strategies to increase fluency and decrease redundancy. When the consumer compares the item description with the item image, they would have a better experience if they see the description is consistent with the image and also sends a message that targets customers’ interest. For example, for a little black dress, the description needs to highlight features such as simple, elegant, must-have, slim cut. When customers see the item fits their needs, they would have a higher chance of adding it to the shopping cart, eventually turning into an actual checkout. Implementing the text generator model can also reduce the research and revision time needed by the content creator and the likelihood of inconsistent descriptions. Overall, our smart text generation model in the e-commerce industry has many advantages over traditional methods as it solves many challenges the platform encounters.

## **9.3 Assessment(ab testing)**

Our hypothesis is that by implementing the text generation model, the customers would have a better engagement and eventually lead to a higher conversion rate in sales. We would use app analytics and tracking analysis to find out how visitors use the site. We would also look at the average time consumers spend on looking at the description, click-through rates as well as conversion rate. We would also conduct quantitative research through interviews and surveys to understand their current opinion on the item description. For an effective campaign, we would see a longer viewing time on the description, a lower bounce rate, and lower cart abandonment.

# **10. Next steps**

## **10.1 Dataset Improvement**

Among the texts generated by the model, we found that there are many similar terms. We checked some mature similar projects of large companies on the Internet and found that the total number of texts supported by their models usually exceeds 1 million, while we only have about 30,000 at this stage. Because of this gap, the diversity of our generated results is lacking, so we need to make more enhancements to the data next. The first point is that we can use a mature machine translation model to translate Chinese text into a foreign language and then translate it back into Chinese, from which we can get new samples with similar semantics. The second point is that we can use OCR technology to batch collect existing product information and text information on the web to build a new dataset. After the data volume is expanded and a better performing model is trained, we can build a semi-supervised model and use the trained model to generate a new source for the reference in our original sample, and continue to train our model as a new sample.

## **10.2 Model Improvement**

The input part of the current model still has a portion of text keywords that need to be entered by humans. We hope to build a model in the future to automatically generate a large number of unordered keywords from images. These keyword sets can be fed into the existing NLP model as input to generate quality marketing text. This is also a way to leverage images at the same time.

This multimodal design combining images and text is academically proven and has been industrialized by some large companies with more resources, so we believe it is feasible. Therefore, in the future, we can use more efficient computational resources to improve the training performance of multimodal models, and add R-CNN modules to capture local information of images.

## **10.3 Final thoughts**

Many times a company's resources are limited, and this is when the development of an AI project needs to be integrated with the company's situation. It is critical to maximize the achievement of mission goals while making the best use of all resources. As in our project, due to unsupported computational resources, more complex models do not work well and often increase the time required for users to use them. In this case, we should always consider the user's situation in order to recommend the most suitable solution.