

Selection and Generation: Learning towards Multi-Product Advertisement Post Generation

Zhangming Chan^{1,2,*}, Yuchi Zhang³, Xiuying Chen^{1,2}, Shen Gao²,
Zhiqiang Zhang³, Dongyan Zhao^{1,2} and Rui Yan^{1,2,†}

¹ Center for Data Science, AAIS, Peking University, Beijing, China

² Wangxuan Institute of Computer Technology, Peking University, Beijing, China

³ Alibaba Group, Beijing, China

{zhangming.chan, xy-chen, shengao, zhaody, ruiyan}@pku.edu.cn
{yuchi.zyc, zhang.zhiqiang}@alibaba-inc.com

Abstract

As the E-commerce thrives, high-quality on-line advertising copywriting has attracted more and more attention. Different from the advertising copywriting for a single product, an advertisement (AD) post includes an attractive topic that meets the customer needs and description copywriting about several products under its topic. A good AD post can highlight the characteristics of each product, thus helps customers make a good choice among candidate products. Hence, multi-product AD post generation is meaningful and important. We propose a novel end-to-end model named S-MG Net to generate the AD post. Targeted at such a challenging real-world problem, we split the AD post generation task into two subprocesses: (1) select a set of products via the SelectNet (Selection Network). (2) generate a post including selected products via the MGenNet (Multi-Generator Network). Concretely, SelectNet first captures the post topic and the relationship among the products to output the representative products. Then, MGenNet generates the description copywriting of each product. Experiments conducted on a large-scale real-world AD post dataset demonstrate that our proposed model achieves impressive performance in terms of both automatic metrics as well as human evaluations.

1 Introduction

With the fantastic development of the Internet, e-commerce has a rapid development and changes the customary shopping way of most people in the world. Advertising copywriting is a huge factor in e-commerce, and well-written advertising copywriting can encourage consumers to understand further and purchase products. However, there is

* This work was done while Z. Chan was an intern at Alibaba Group. Y. Zhang works at Ant Group now.

† Corresponding Author: Rui Yan (ruiyan@pku.edu.cn).

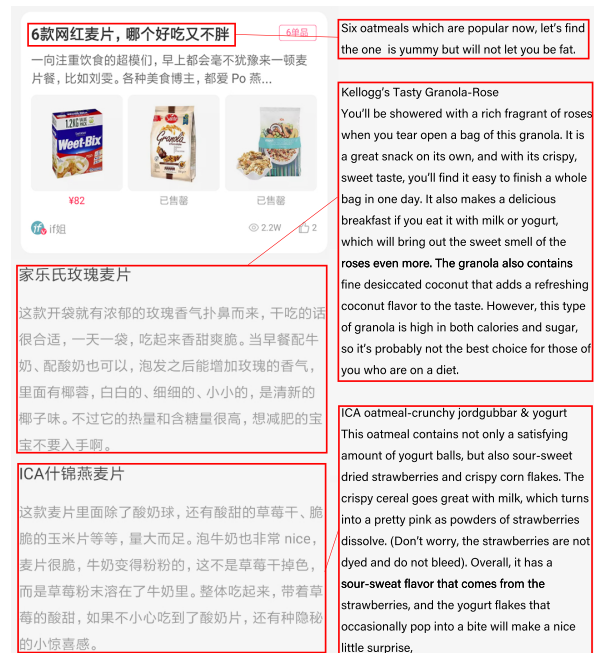


Figure 1: A case of multi-product AD post (with translation) from Taobao. Due to space limitation, we only show the post title and the first two copywriting.

an important restriction factor for traditional advertising copywriting, i.e., the writing efficiency of human copywriters cannot match the growth rate of new products. Many e-commerce websites, such as Amazon¹, Taobao² and JD³, have billions of products, so it is impossible to write all copywriting manually. To address this issue, researchers pay more and more attention to the automatic advertising copywriting generation. The initial work (Wang et al., 2017) on automatic advertising copywriting generation is based on the predefined template. With the surge of deep learning techniques, the neural encoder-decoder framework shows remarkable effects on various text generation tasks, e.g., sum-

¹<https://www.amazon.com/>

²<https://www.taobao.com/>

³<https://www.jd.com/>

marization (See et al., 2017; Chen et al., 2019b; Gao et al., 2019a, 2020), story generation (Fan et al., 2018; Li et al., 2019; Yao et al., 2019) and so on. The researchers from academia and industry begin to explore how to generate product advertising copywriting through deep learning methods. Zhang et al. (2019) propose a pointer-generator model to generate the product advertising copywriting whose patterns are controlled. Chen et al. (2019a) explore a new way to generate personalized product copywriting by enhancing the transformer with an extra knowledge base.

However, all the previous studies focus on copywriting generation for a single product. In such case, consumers need to compare among products and summarize the advantages and disadvantages of each product by themselves. This makes online shopping a challenging and time-consuming task for customers with high requirements for products. Consequently, a novel advertising copywriting format called multi-product AD post is becoming increasingly popular these years. A multi-product AD post contains several related products that either function similarly or match each other. Each product in the post has its own copywriting, and every copywriting will take full account of the topic of the post and other products' information. The characteristic information of each product can be highlighted by considering both the post topic and other products' information, so the post can let customers understand the characteristics of each product quickly. We show a case of AD post in Figure 1 and we can find the keyword "fat" in the post title and it's the reason for the first product copywriting to emphasize "high-calorie" attribute of the products. Meanwhile, besides the most basic information about cereal, copywriting about these two products emphasize the unique features, such as "rose" for the first product and "yogurt" for the second. However, the post writing is far more difficult than single product copywriting. First, it needs to select suitable products for writing in a post. Then the copywriter must consider the relationship between the post topic and the products to describe each product clearly. Meanwhile, to describe the unique characteristics of each product, the copywriter should also consider the information of other products in the post. Therefore, the multi-product AD post can only be used on a few hot products yet because of the low writing speed and the high cost. Thus, it is very necessary to realize the automatic

generation of multi-product AD post.

Targeted to automatic multi-product AD post generation, we propose a two-stage model named S-MG Net. In this model, we split the multi-product AD post generation task into two subprocesses: (1) select a set of products via the SelectNet (Selection Network). (2) generate a post including selected products via the MGenNet (Multi-Generator Network). Concretely, we first propose an iterative attention mechanism called Select-Attention (SelectAttn) and build a model called SelectNet. SelectAttn allows the SelectNet to select the appropriate combination of the products based on the post topic and the relationship among the products. Then, we design the MGenNet inspired by the multi-agent communication method to generate each product's copywriting considering the topic and other products. In detail, we let each agent, i.e., the generator from MGenNet, generate product copywriting individually. Meanwhile, we propose an agent communication strategy that can let each agent obtain the information of other products when the product copywriting is generating. Finally, we combine all the generated copywriting to form the whole multi-product AD post.

In a nutshell, our contributions can be summarized as: (1) We propose the SelectNet which can select the appropriate combination of the products based on the post topic and the relationship among the products. (2) We build the MGenNet to generate the copywriting of each product based on the multi-agent communication framework. (3) We combine the above networks as an end-to-end S-MG Net which can generate attractive AD posts.

2 Related Work

Text generation. Text generation has become one of the hottest subfields of natural language processing. Previous researches mainly focus on several popular text generation tasks, such as dialogue generation (Serban et al., 2016; Tao et al., 2018a; Hu et al., 2019; Chan et al., 2019b) and story generation (Xu et al., 2018; Li et al., 2019; Yao et al., 2019). Bowman et al. (2016) are proposed to improve wording novelty. Serban et al. (2017) target to the intra-sentence consistency and thematic consistency is improved by Fan et al. (2018); Litschko et al. (2018). Besides, text generation from different data formats have also been widely studied in recent years, e.g., table-to-text generation (Liu et al., 2018), which can fit various data formats in

real-world scenarios.

Multi-Agent Communication. CommNet proposed by Sukhbaatar et al. (2016) is the first deep learning framework for multi-agent communication. There are several previous works built on the CommNet framework, for example, researchers use the multi-agent communication method to play the starcraft games (Vinyals et al., 2017). Mordatch and Abbeel (2018) deal with natural language processing tasks such as machine translation and sentiment analysis with the multi-agent communication method. Celikyilmaz et al. (2018) present the first study using the multi-agent framework for summarization.

Product descriptions generation. Product description copywriting is critical for the e-commerce platform, and automatically generating the product description copywriting has attracted considerable interest from both academia and industry because of its importance. Wang et al. (2017) first focus on the product description generation task and incorporates the preset template to generate product descriptions automatically. With the development of deep learning, Zhang et al. (2019) proposed a pointer-based generation model with a dual encoder to generate product description and achieved the controlled patterns. Chen et al. (2019a) proposed a transformer-based model to generate personalized product descriptions by combining a knowledge graph as the extra knowledge base. Chan et al. (2019a) proposed use the entity label to enhance the fidelity of the product description.

3 Problem Formulation

To formulate the multi-product AD post generation task, we use P to denote a product candidate set which contains a lot of productions, namely $P = \{u_1^p, u_2^p, \dots, u_{n_p}^p\}$, where u_i^p indicates the information⁴ of i -th product in the set and n_p is the set size. Specifically, $u_i^p = \{w_{i,1}^p, w_{i,2}^p, \dots, w_{i,n_{p,i}}^p\}$ is a text sequence which contains $n_{p,i}$ words. Meanwhile, the topic of the AD post is represented as $T = \{w_1^t, w_2^t, \dots, w_{n_t}^t\}$ where w_i^t indicates the i -th word in the topic sequence and n_t is the length. The goal of our model is to generate a multi-product AD post $\hat{C} = \{\hat{u}_1^c, \hat{u}_2^c, \dots, \hat{u}_{n_c}^c\}$ where $\hat{u}_i^c = \{\hat{w}_{i,1}^c, \hat{w}_{i,2}^c, \dots, \hat{w}_{i,n_{c,i}}^c\}$ represents the copywriting of the i -th product in the post which

⁴In this work, the product information comes from the product title which contains the attributes keywords of the product. Each product title can be regarded as a text sequence.

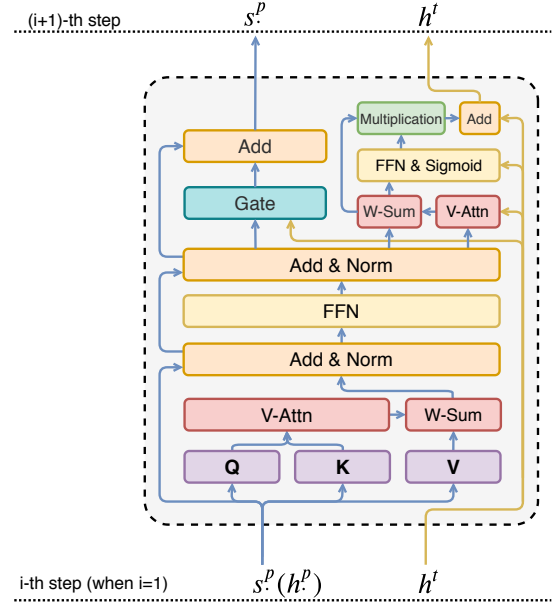


Figure 2: The SelectAttn mechanism. The V-Attn, W-Sum indicate the weight calculation and weighted sum respectively. The FFN indicates the Feed Forward Network. The first step input is h^p , and others are the s^p .

contains $n_{c,i}$ words. Essentially, our model optimizes the parameters to maximize the probability $p(C|P, T)$ where C is the ground truth post. Note that we ignore the order of products in the post and regard each product as equal.

4 Proposed Model

As Figure 3 shows, our proposed S-MG Net model consists of Selection Network (SelectNet) and Multi-Generator Network (MGenNet).

4.1 Selection Network

We propose the Selection Network (SelectNet) to select the appropriate products from the input product candidate set. Inspired by self-attention mechanism proposed by Vaswani et al. (2017), we design the Select-Attention (SelectAttn) mechanism to capture the relationship among the products and the relationship between the post topic and each product. Note that the SelectAttn will iterate N time, we focus on one of these iterations for clarity. Figure 2 shows the structure of SelectAttn.

To begin with, we obtain the post topic T and the product candidate set P . More details of these symbols are mentioned in Section 3. First, we use an embedding matrix e to embed each word in T and P into a high-dimensional vector space. Then, we use an RNN encoder named topic encoder to encode the embedded post topic $e(T)$ to

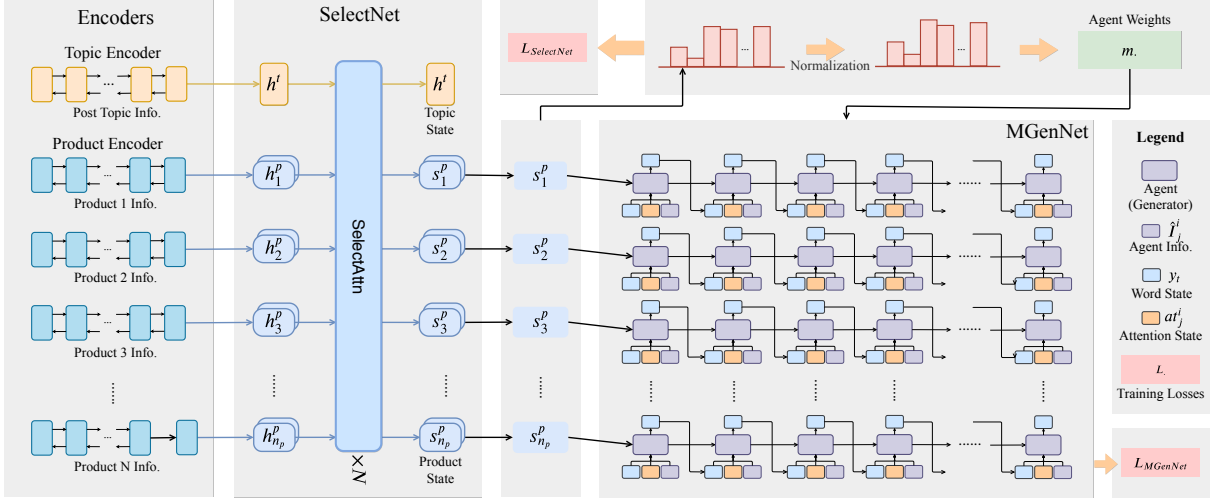


Figure 3: The overview of the end-to-end S-MG Net which is the combination of SelectNet and MGenNet model.

h^t . Similarly, we use another RNN encoder called product encoder to encode each embedded product information sequence $e(u_i^p)$ as the h_i^p , so we can obtain $h^P = \{h_1^p, h_2^p, \dots, h_{n_p}^p\}$ which indicate all the product candidates.

To capture the relationship among the product candidates, we use the self-attention mechanism to conduct the interaction between $\{h_1^p, h_2^p, \dots, h_{n_p}^p\}$. Concretely, we use three fully-connected layers to project h_i^p into three spaces, i.e., $Q_i = \text{FC}_q(h_i^p)$, $K_i = \text{FC}_k(h_i^p)$, $V_i = \text{FC}_v(h_i^p)$, where Q_i , K_i and V_i represent the query, the key and the value respectively. The attention module takes Q_i to attend to each K_j ⁵, and uses these attention distribution results $\alpha_{i,j} \in \mathbb{R}^N$ as the weights to obtain the weighted sum of v_j :

$$\alpha_{i,j} = \frac{\exp(Q_i K_j)}{\sum_{n=1}^{n_p} \exp(Q_i K_n)}, \quad (1)$$

$$\beta_i = \sum_{j=1}^{n_p} \alpha_{i,j} v_j,$$

where $\alpha_{i,j}$ denotes the attention weight of the i -th product, i.e., h_i^p , on the j -th product, i.e., h_j^p . Next, we add the original product representation h_i^p on β_i as the residual connection layer, as shown in:

$$\hat{s}_i^p = \text{LayerNorm}(h_i^p + \beta_i), \quad (2)$$

where LayerNorm indicates the Layer Normalization. Next, we apply a feed-forward layer on \hat{s}_i^p to obtain the representation \hat{s}_i^p and conduct the residual connection layer again:

$$\begin{aligned} \hat{s}_i^p &= \text{ReLU}(\hat{s}_i^p \cdot W_1 + b_1) \cdot W_2 + b_2, \\ \hat{s}_i^p &= \text{LayerNorm}(\hat{s}_i^p + \hat{s}_i^p), \end{aligned} \quad (3)$$

⁵We use the subscript “.” to represent the any index, for example, K . represents K_1, K_2, \dots, K_N .

where W_1, W_2, b_1, b_2 are all trainable parameters. Then the normalization operation is applied as introduced before. The above operation can model the relationship among the products.

Next, to target at utilizing the relationship between post topic and each product, we propose to integrate the information of post topic into each product presentation \hat{s}_i^p as additional prior information. We use a gate mechanism to control the fusion between topic information and each product representation and we add the product in the representation as the residual connection layer:

$$\begin{aligned} g^p &= \text{Sigmoid}([\hat{s}_i^p; h^t] \cdot W_g + b_g) \\ \tilde{s}^p &= g^p \cdot h^t + (1 - g^p) \cdot \hat{s}_i^p, \\ s^p &= \hat{s}_i^p + \tilde{s}^p, \end{aligned} \quad (4)$$

where W_g, b_g are trainable parameter matrices. After adding post topic information to product representation, we use the raw product information to polish the topic representation. Concretely, we use the cross attention mechanism to obtain the information that we used to polish the topic representation h^t , the process is shown as below:

$$\begin{aligned} \gamma_i &= \frac{\exp(\hat{s}_i^p h^t)}{\sum_{j=1}^{n_p} \exp(\hat{s}_j^p h^t)}, \\ \delta &= \sum_{i=1}^{n_p} \gamma_i \hat{s}_i^p. \end{aligned} \quad (5)$$

Inspired by GLU (Gehring et al., 2017), we use the product and topic information to control the amount of information of the δ . Finally, we add the δ to h^t to update the topic information as following:

$$\begin{aligned} g^t &= \text{Sigmoid}([\delta; h^t] \cdot W_t + b_t), \\ \tilde{\delta} &= g^t \cdot \delta, \\ h^t &= h^t + \tilde{\delta} \end{aligned} \quad (6)$$

where W_t, b_t are all trainable parameters and h^t has been updated by the product information. We name the whole above operation as SelectAttn. Finally, we obtain the new product representations $s^P = \{s_1^p, s_2^p, s_3^p, \dots, s_{n_p}^p\}$.

Lastly, we use these product representations from SelectAttn to select the products which can be shown in the AD post. After interacting with the post topic and other products, the product representations can describe the matching degree of the product with the target post. So the selection is based on the scores calculated from product representations. We can obtain the scores from below:

$$score. = \text{Sigmoid}(s^p \cdot W_c), \quad (7)$$

where W_c represents a trainable matrix. Finally, we can obtain a group of probability samples, i.e., $\{score_1, score_2, \dots, score_{n_p}\}$ as the final score. We use the scores to rank the products and use the top- M products as the selection result.

4.2 Multi-Generator Network

After we finish the selection process, we continue to generate the copywriting for each selected product. Inspired by Celikyilmaz et al. (2018), we propose to build Multi-Generator Network (MGenNet) based on the multi-agent framework and a generator is acting as an agent. In the following, we use agent to indicate the generator for more intuitive explanations.

As mentioned before, we rank the selection scores $\{score_1, score_2, \dots, score_{n_p}\}$ and get the top- M products as selection result $\bar{s}^p = \{\bar{s}_1^p, \bar{s}_2^p, \bar{s}_3^p, \dots, \bar{s}_M^p\}$ where $\{\hat{m}_1, \hat{m}_2, \hat{m}_3, \dots, \hat{m}_M\}$ is the corresponding scores. We normalize these scores as:

$$m. = \frac{\hat{m}. - \text{Min}(\hat{m}.)}{\text{Max}(\hat{m}.) - \text{Min}(\hat{m}.)}, \quad (8)$$

where Min and Max indicate the operations which obtain the minimum and maximum value from the score set. Finally, we obtain $\{m_1, m_2, m_3, \dots, m_M\}$ where $m.$ is a value in $[0, 1]$. We regard m_i as the importance of the i -th product in the AD post and use it as the weight of the agent which generates the copywriting of the i -th product.

We define each RNN cell as an agent and all RNN cells share the same parameters in this work for promoting the generation efficiency. We let every agent generate copywriting for one product individually. To initialize the each agent, we use

a linear layer to cover all the selected product representation $\{\bar{s}_1^p, \bar{s}_2^p, \bar{s}_3^p, \dots, \bar{s}_M^p\}$ to initialize the corresponding agent (RNN Cell) and the use a special token “<BOS>” as the input of the first step, which can be described as follows:

$$\begin{aligned} a_0 &= \text{ReLU}(\bar{s}^p \cdot W_s) + b_s, \\ a_1 &= \text{Agent}(a_0, e(\langle \text{BOS} \rangle)), \end{aligned} \quad (9)$$

where Agent indicates the operation of RNN and a_i represents the state of agent in i -th step. During the generation, we let the agents communicate with each other. In order to deliver useful information, we collect the status of all agents except themselves. Meanwhile, we use the agent weight $m.$ coming from Equation 8 to control the weight of the status. The process is as follows:

$$\begin{aligned} \hat{a}_j &= a_j \cdot m., \\ I_j^i &= \{\hat{a}_j^1, \dots, \hat{a}_j^{i-1}, \hat{a}_j^{i+1}, \dots, \hat{a}_j^M\} \end{aligned} \quad (10)$$

where \hat{a}_j^i represents the information coming from the i -th agent when j -th step and I_j^i represents the information set for i -th agent when j -th step.

Inspired by Celikyilmaz et al. (2018) and Sukhbaatar et al. (2016), we propose a multi-agent communication strategy to calculate practical information that the agent obtains from other agents in the generation process. We use the mean value of I_j^i shown in Equation 10 as the practical information for the i -th agent in the j -th step as:

$$\hat{I}_j^i = \frac{\text{sum}\{I_j^i\}}{M-1}. \quad (11)$$

The \hat{I}_j^i contains all the information of other agents, and the i -th agent can get more information as the prior for generating.

After obtaining the information \hat{I}_j^i , we attach it as an extra input to the corresponding agent. We use the below equation to express this process

$$\begin{aligned} y'_t &= ([y_t; \hat{I}_j^i; a_t] \cdot W_y) + b_y, \\ a_{t+1} &= \text{Agent}(a_t, y'_t), \end{aligned} \quad (12)$$

where W_y, b_y are trainable parameters. y_t is the input of agent at t -th time step and the a_t is the attention vector which is calculated from the corresponding product RNN encoder status as same as Luong et al. (2015). Then, we use a linear layer to obtain the generated word. Finally, we can use the beam search algorithm to get all copywriting $\hat{C} = \{\hat{u}_1^d, \hat{u}_2^d, \dots, \hat{u}_M^d\}$ which $\hat{u}_i^d = \{\hat{w}_{i,1}^d, \hat{w}_{i,2}^d, \dots, \hat{w}_{i,L_{c,i}}^d\}$.

4.3 Training Objective

To start with, we combine SelectNet and MGenNet as an end-to-end framework. We launch the following objective to minimize the MLE loss between the ground truth and generated copywriting. Meanwhile, we minimize the loss between selected product ground truth in real multi-product AD post, and the objective function is:

$$\mathbb{L} = \lambda \mathbf{E}_\theta \log p(\hat{P}|P, T) + \gamma \mathbf{E}_\phi \log p(\hat{C}|\hat{P}, T), \quad (13)$$

where λ , γ are the weights for SelectNet and MGenNet loss respectively. \hat{P} indicates the selected product and \hat{C} indicates the generated post.

5 Experiments

5.1 Dataset

We construct a multi-product AD post dataset collected from Bimai Qingdan (which means a list of goods you must buy) in Taobao. Millions of posts are composed by professional copywriters to introduce and recommend different products for online shoppers. Each post consists of several products with their images, description copywriting, and a title that reflects the topic of this post.

We extract the title of each post as the post topic T . To construct product candidate set P , we sample negative samples from the product set which have the same categories (such as food, clothes and so on.) with the positive samples, i.e., the products in the real post and the ratio of positive examples and negative examples is 6:9. After we select the products for the candidate set, we obtain the corresponding short product attributes title and regard them as product information $\{u_1^p, u_2^p, \dots, u_{n_p}^p\}$. Naturally, the long product description copywriting of all products in the post are regarded as the target. As a result, we obtain 339,433 posts and randomly split them into 330,000 / 4,433 / 5,000 as training / validation / testing sets.

5.2 Hyperparameters.

The dimension of word embeddings is set to 256. The N of the SelectAttn in the SelectNet is set to 3. The vocabulary comprises the most frequent 50,000 words and we let the encoder and decoder share this common vocabulary. The product encoder and topic encoder in the SelectNet model are two bi-directional RNN with the LSTM cells, respectively. The agent decoder in MGenNet consists of a one-layer RNN with LSTMs. The hidden state sizes of both LSTM in encoder and decoder are set to

Table 1: Criteria of human evaluation.

Readability	Is the copywriting grammatically formed and smooth?
Informativeness	Does the copywriting contains informative words?
Attractiveness	How attractive the post copywriting is?
Rationality	Is the product selection reasonable?

512. We set the size of mini-batch to 64 and all sequence inputs were padded with the special token “<PAD>” to a maximum sequence length of the batch. λ is set to 2 in the training process. During decoding, we employ the beam search with a beam size of 4 to generate a more fluent sentence. We train all models for 40 epoch and it took around 20 hours on GPUs for training. After each epoch, we evaluated our model on the validation set and chose the best performing model for testing. We use the Adam optimizer (Duchi et al., 2010) for all experiments and the learning rate is set to 1e-3.

5.3 Baselines

To show the effectiveness of our model, several classical generation approaches are set as baselines. 1). **Seq2Seq** (Bahdanau et al., 2014). A classic and widely used text generation model. 2). **ConvSeq** (Gehring et al., 2017). A model combining the CNN and the Seq2Seq network. 3). **Transformer** (Vaswani et al., 2017). The state-of-the-art model in several text generation tasks. 4). **PCPG** (Zhang et al., 2019). A pattern-controlled product description generation model. We adapt it to our scenario. 5). **KOBE** (Chen et al., 2019a). A knowledge-based and personalized product description generation model based on Transformer. We adapt the model for our scenario by removing the personal module.

We cannot generate copywriting for each product individually when we don’t know the corresponding relation. To apply these baseline models to our scenario, we separate the description text of each product in the post with a unique character “<SOP>” and then concatenate the whole text as the final target for training.

To analyze our SelectNet, we select several models for text matching as baselines. 1). **RNN** (Liu et al., 2016). A model uses RNN to model input as the hidden state to classify. 2). **SCN** (Wu et al., 2019). A classic retrieval model for response selection. The model lets the post title interact with product information and transforms interaction matrices into a matching vector with CNN. 3). **MRFN** (Tao et al., 2019). A strong retrieval-

Table 2: The automatic metric result of our model and baselines. Since the number of sentences generated by ConvSeq is extremely small, so evaluating the intra-dist metric is meaningless, and we omit the intra-dist score. The results of our methodologies are significant with p-value <0.05 measured by t-test over the best baseline.

Models	Embedding Metrics			Inter-Distinct				Intra-Distinct			
	Average	Greedy	Extrema	Dist-1	Dist-2	Dist-3	Dist-4	Dist-1	Dist-2	Dist-3	Dist-4
Seq2seq	0.9197	548.69	0.4293	0.937	3.183	6.655	9.940	20.77	28.38	31.07	33.41
ConvSeq	0.6049	326.99	0.1123	1.308	2.796	3.970	4.951	-	-	-	-
Transformer	0.8662	537.69	0.3941	1.473	4.427	9.099	13.59	25.23	33.83	39.79	45.48
PCPG	0.8830	540.41	0.3713	1.409	3.943	7.423	10.43	22.43	29.98	36.80	40.31
KOBE	0.8783	539.23	0.4023	1.523	5.334	11.34	18.32	26.46	37.43	43.23	53.84
S-MG_C	0.9438	560.45	0.4481	1.763	8.051	18.37	28.30	44.66	66.22	73.57	78.96
S-SG	0.8774	566.86	0.4280	1.294	4.059	8.479	12.71	24.76	33.06	38.72	44.09
S-MG	0.9428	558.62	0.4440	1.713	7.502	17.21	26.60	44.49	65.97	73.26	78.64

based model in response selection. Tao et al. (2019) encode the interaction between two texts from multiple kinds of representations and study how to fuse them. 4). **SETM** (Yang et al., 2019). It lets the post title interact with product text and uses the representation to obtain a weight matrix for each word. 5). **Self-Attn** (Vaswani et al., 2017). We use the self-attention mechanism to capture the relationship among the product candidates for product selection.

We also conduct the exploration of the MGenNet, and the setting is shown as follows 1). **S-MG_C**. Our proposed multi-agent generation framework with communication strategy. 2). **S-MG**. The original multi-agent generation framework without communication. 3). **S-SG**. We replace the MGenNet with a pretrained Transformer decoder and generate the whole post copywriting in one pass.

5.4 Evaluation Metrics

As mentioned in Section 3, we ignore the order of products in the post and regard each product as equal. Hence, traditional metrics such as BLEU and ROUGE, are unsuitable in this scenario. To evaluate the results of the generated AD post, we adopt the following widely used metrics.

Embedding Metrics. To obtain semantic matches between the generated copywriting and ground-truth, we perform evaluation using the embedding metrics. Following Gao et al. (2019b), we calculate three measures: 1) *Average*, cosine similarity between the averaged word embeddings in the two utterances (Mitchell and Lapata, 2008); 2) *Greedy*, i.e., greedily matching words in two utterances based on the cosine similarities, and the

total scores are then averaged across all words (Rus and Lintean, 2012); 3) *Extrema*, cosine similarity between the largest values among the word embeddings in the two utterances (Forgues et al., 2014). The used word2vec embedding is trained by ourselves because there is no open-access e-commerce embedding.

Distinct Metrics. We use distinct scores to reflect the diversity of the copywriting. Dist- n is defined as the ratio of unique n -grams ($n = 1, 2, 3, 4$) overall n -grams in the generated copywriting. Following Gu et al. (2018), we define *intra-dist* as the average of distinct values within each copywriting and *inter-dist* as the distinct value among all copywriting.

Human Evaluation. Because Tao et al. (2018b) mentioned that only using the automatic metrics to evaluate text quality can be misleading, we also conduct human evaluation. Three well-educated annotators are hired to evaluate the quality of generation, where the evaluation is conducted in a double-blind fashion. 200 randomly sampled copywriting generated by each model are rated by each annotator with four different aspects. Details of the criteria are illustrated in Table 1. All criteria are scored from 1 to 4, i.e., from bad to good.

6 Results and Analysis

6.1 Overall Performance

We report the performance of our proposed S-MG Net and baselines in terms of all the automatic metrics, and the results are shown in Table 2. Firstly, among all baselines, we find the KOBE model can obtain the best performance on the distinct metrics, and the Seq2seq model performs best on the em-

Table 3: The result of our proposed SelectNet model and baselines. Rank is the accuracy in Top-6 prediction compare with the ground-truth. The p-value of our results is smaller than 0.01 measured by t-test over the best baseline.

Model	Valid Set		Test Set	
	Acc	Rank	Acc	Rank
RNN	61.77	46.89	61.75	46.84
SCN	70.92	61.48	70.91	61.49
MRFN	75.45	71.32	75.23	71.62
SETM	71.22	61.90	69.53	59.47
Self-Attn	88.59	84.81	88.60	84.83
SelectNet	90.22	86.88	90.22	86.89

bedding metrics. In all, our proposed S-MG Net outperforms all baselines on almost all metrics, outperforming the strongest baseline KOBE in distinct metrics, by 15.76%, 50.94%, 68.78%, and 76.92% in terms of Inter-Dist-1, Inter-Dist-2, Intra-Dist-1, and Intra-Dist-2, respectively. Besides, S-MG Net outperforms the strongest baseline Seq2seq model in embedding metrics by 2.62%, 2.14% and 4.38% on *Average*, *Greedy* and *Extrema* score. Meanwhile, our model also performs well on human evaluation. As shown in the Table 4, S-MG Net obtains 6.73%, 17.68%, 13.09% and 10.93% improvement than the KOBE model in terms of readability, Informativeness, attractiveness and rationality scores. Overall, our proposed model brings an impressive improvement.

6.2 Discussions

Comparing with all baselines, our proposed SelectNet model achieves an impressive improvement. As shown in Table 3, through conducting the comparison between our SelectNet model and all baselines, we can find our SelectNet shows an impressive improvement than all baselines. For example, SelectNet outperforms the MRFN model on the testing set by 19.93% and 21.32% on the Test Acc score and Test Rank score. The improvement is very impressive.

The relationship between the products is crucial. Meanwhile, the relationship between post title and each product is also important for the improvement. As shown in Table 3, Self-Attn which captures the relationship among the products outperforms the strongest MRFN model by 17.77% and 18.44% on Acc score and Rank score on the testing set. According to such impressive improve-

Table 4: Human evaluation with the strongest baselines. The Read., Info., Attract., Ration. indicate the Readability, Informativeness, Attractiveness and Rationality, respectively. The mean kappa value between annotators is 0.43.

Model	Read.	Info.	Attract.	Ration.
GroundTruth	3.982	3.812	3.882	3.902
Seq2seq	3.100	2.690	2.435	1.895
KOBE	3.415	2.885	3.095	2.835
S-MG_C	3.645	3.395	3.500	3.145

ment, we can learn the interaction between the products is crucial in this product selection process. Meanwhile, by comparing the Self-Attn model and our SelectNet model, we can find our SelectNet model which uses the relationship between post topic and each product can outperform the Self-Attn model by 1.62 and 2.06 in the Test Acc score and Test Rank score. It is clear that the interaction between post topic information and each product can improve the effects of the product selection.

Multi-agent generation brings a leap improvement. As we mentioned before, multi-agent generation framework is different from the traditional encoder-decoder framework which only contains one single decoder. As shown in Table 2, S-MG achieves a further improvement over all the encoder-decoder framework baselines. Specifically, S-MG wins over the best performing Seq2seq and KOBE model in the baselines. S-MG in the comparison of metrics, on the *Average*, *Greedy*, *Extrema* with the Seq2seq model, obtains improvements as 2.51%, 1.81%, 3.42%. Compare with the KOBE model, S-MG outperforms by 12.48% and 68.14% in the Inter-Dist-1 and Intra-Dist-1, respectively. Compared with the S-SG which combines the SelectNet with a single decoder, S-MG can also achieve 7.45% (*Average*), 3.74% (*Extrema*), 33.52% (Inter-Dist-1) and 79.68% (Intra-Dist-1) improvements. These impressive improvements show that S-MG Net outperforms the traditional encoder-decoder framework in the multi-product AD post generation task. Meanwhile, as the ablation experiment about the agent communication, we compare the S-MG_C with the S-MG, and we can find the S-MG_C outperform the S-MG in all metrics. According to the above analysis, we can learn that agent communication can improve the generation performance further.

Table 5: A generated copywriting. The (✓) represent this product is in the groundtruth and the (×) is the opposite.

Title	赶快解锁女装利器，从此迎接美好生活(Unlock the weapon of women’s clothes, and welcome the beautiful life from now on)
Product Set	<p>(×) 1. d’zzit 地素夏专柜新款运动风半身裙(D’zzit’s new sporty style skirt which is designed for the summer.)</p> <p>(✓) 2. vifille 同款刺绣撞色卫衣 (This hoodie has vibrant color and embroidery and it is so fashione that Vifille also wear it !)</p> <p>(✓) 3. beanpole 滨波女士宽松版印花卫衣(BEAN POLE’s women hoodie with printing and it is loose for wearing.)</p> <p>(×) 4. 2016 夏季新款女装名媛条纹不规则假两件开叉露肩连衣裙夏中长款 (Fake two off-the-shoulder split women dresses with irregular stripes, and it is the special style in 2016)</p> <p>(×) 5. 优雅七分袖夏季宽松a字裙(It is an elegant summer A-line skirt with three-quarter sleeves.)</p> <p>(×) 6. 白夜宽松蝙蝠袖套头毛衣(White-night’s bat-sleeved sweater, and it is loose for wearing.)</p> <p>(✓) 7. 果酱公主春新原创森系时尚气质卫衣 (Guojiang Princess’s original fashion hoodie for the spring and you will be very comfortable because it is like fresh air.)</p> <p>(×) 8. 梦舒雅冬新款加厚加绒微喇裤修身(Mengshuaya’s new thickened pants and it has fleecy trim.)</p> <p>(✓) 9. 七格格卫衣加绒半高领秋中长款宽松 (Qigege’s hoodie with fleece trim and semi-high collar. It is the middle length for autumn. It is loose for wearing.)</p> <p>(×) 10. 休闲七分显瘦通勤正装裤(7-points pants for relaxing, it will make you look more thin.)</p> <p>(×) 11. yiner 冬时尚贴布侧条纹系带连衣裙(Yiner’s fashion dress with side stripes and straps, it is suitable for winter.)</p> <p>(×) 12. loftshine 塔炫女装专柜大衣(Loftshine’s clothing coat, it is designed for women.)</p> <p>(×) 13. 洋气短袖碎花小清新超仙气质雪纺衫(This skirt which has short sleeve and the broken flower is pure and fresh.)</p> <p>(✓) 14. 卫衣女nina mist 套头飘带(This hoodie is designed for women by Nina Mist’s and it has a beautiful ribbon.)</p> <p>(✓) 15. 韩版学生蝙蝠袖连帽卫衣(Korean style hoodie with bat sleeve. It is suitable for students.)</p>
S-MGC	<ul style="list-style-type: none"> • [For Product 14]: 这款卫衣采用了经典的圆领设计，简约大方，宽松的版型 (This hoodie adopts the classic round collar design, it is simple and generous. Meanwhile, its version is loose.) • [For Product 7]: 这款卫衣采用了柔软的针织面料，上身舒适又保暖 (This hoodie is soft and like air, and it keeps the body warm and comfortable.) • [For Product 15]: 韩版风格，充满优雅气息，版型宽松，上身就是时髦 (The hoodie is Korean style and elegant. Its loose version makes owner fashion.) • [For Product 9]: 中长款的长度，百搭不挑身形，版型宽松，上身就是时髦 (The hoodie is middle length and it is suitable for multiple body shapes. Its loose version makes owner fashion.) • [For Product 2]: 精致的刺绣，点缀着整体的时尚不呆板，经典的圆领 (This clothes has delicate embroidery which shows the fashion, and it has classic round collar.) • [For Product 3]: 这是来自BEAN POLE 的卫衣，适合女士在休闲场合穿着 (This is a hoodie from BEAN POLE and it is suitable for casual wearing.)

6.3 Case Study

Table 5 shows the generated multi-product AD post of our proposed S-MG Net model. In the product candidate set, almost all products are related to women’s clothes and it is hard for SelectNet to conduct the product selection only depends on the topic information. In this case, our SelectNet selects a product set consisting of 6 hoodies and it is categorical that these 6 hoodies can match each other. It proves that the relationship between the products is crucial in our SelectAttn mechanism. Meanwhile, our model can generate an informative and attractive multi-product AD post. The copywriting for each product is unique and informative.

7 Conclusion and Future Work

In this paper, we explore the multi-product AD post generation which is meaningful for both academic researches and industrial applications. Due to the characteristics of such a new task, most existing classical text generation schemes do not work well. To fill this gap, we propose an end-to-end S-MG Net model that uses the SelectNet to select the combination of products associated with

each other and uses the MGenNet to generate the copywriting for each product. After stitching the generated copywriting together, we can obtain a well-formed multi-product AD post. Experiments conducted a large-scale real-world product description dataset demonstrate that our proposed model achieves promising and impressive performance.

Owing to the promising results of the proposed two-stage model, we could degenerate this process into more specific sub-processes. Moreover, we will conduct further exploration of the multi-product AD post form, including more vivid multimedia information, such as pictures and videos.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. We would also like to thank Yongliang Wang from Ant Group for his suggestions and help. This work was supported by the National Key Research and Development Program of China (No.2020AAA0105200), and the National Science Foundation of China (NSFC No.61876196, No.61672058). Rui Yan is partially supported as a Young Fellow of Beijing Institute of Artificial Intelligence (BAAI).

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint*.
- Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *SIGNLL*, Berlin, Germany.
- Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. *arXiv preprint*.
- Zhangming Chan, Xiuying Chen, Yongliang Wang, Juntao Li, Zhiqiang Zhang, Kun Gai, Dongyan Zhao, and Rui Yan. 2019a. Stick to the facts: Learning towards a fidelity-oriented e-commerce product description generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4959–4968, Hong Kong, China. Association for Computational Linguistics.
- Zhangming Chan, Juntao Li, Xiaopeng Yang, Xiuying Chen, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019b. Modeling personalization in continuous space for response generation via augmented wasserstein autoencoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1931–1940.
- Qibin Chen, Junyang Lin, Yichang Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019a. Towards knowledge-based personalized product description generation in e-commerce. *arXiv preprint arXiv:1903.12457*.
- Xiuying Chen, Zhangming Chan, Shen Gao, Meng-Hsuan Yu, Dongyan Zhao, and Rui Yan. 2019b. Learning towards abstractive timeline summarization. In *IJCAI*, pages 4939–4945.
- John C. Duchi, Elad Hazan, and Yoram Singer. 2010. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *ACL*, volume 1.
- Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In *Nips, workshop*, volume 2.
- Shen Gao, Xiuying Chen, Piji Li, Zhaochun Ren, Lidong Bing, Dongyan Zhao, and Rui Yan. 2019a. Abstractive text summarization by incorporating reader comments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6399–6406.
- Shen Gao, Xiuying Chen, Zhaochun Ren, Dongyan Zhao, and Rui Yan. 2020. From standard summarization to new tasks and beyond: Summarization with manifold information. In *IJCAI*.
- Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019b. Product-aware answer generation in e-commerce question-answering. In *WSDM*, pages 429–437. ACM.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.
- Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2018. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. *arXiv preprint arXiv:1805.12352*.
- Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. 2019. Gsn: A graph-structured network for multi-party dialogues. *arXiv preprint arXiv:1905.13637*.
- Juntao Li, Lidong Bing, Lisong Qiu, Dongmin Chen, Dongyan Zhao, and Rui Yan. 2019. Learning to write stories with thematic consistency and wording novelty. In *AAAI*, volume 33, pages 1715–1722.
- Robert Litschko, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. 2018. Unsupervised Cross-Lingual Information Retrieval Using Monolingual Data Only. *SIGIR*.
- Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.
- Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *AAAI*.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. *NAACL-HLT*, pages 236–244.
- Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *AAAI*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

- Vasile Rus and Mihai Lintean. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 157–162. ACL.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Association for Computational Linguistics*, pages 1073–1083.
- Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. In *AAAI*.
- Sainbayar Sukhbaatar, Rob Fergus, et al. 2016. Learning multiagent communication with backpropagation. In *NIPS*, pages 2244–2252.
- Chongyang Tao, Shen Gao, Mingyue Shang, Wei Wu, Dongyan Zhao, and Rui Yan. 2018a. Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In *IJCAI*, pages 4418–4424.
- Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018b. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 267–275.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. 2017. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*.
- Jinpeng Wang, Yutai Hou, Jing Liu, Yunbo Cao, and Chin-Yew Lin. 2017. A statistical framework for product description generation. In *IJCAI*, pages 187–192.
- Yu Wu, Wei Wu, Chen Xing, Can Xu, Zhoujun Li, and Ming Zhou. 2019. A sequential matching framework for multi-turn response selection in retrieval-based chatbots. *Computational Linguistics*, 45(1):163–197.
- Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018. A Skeleton-Based Model for Promoting Coherence Among Sentences in Narrative Story Generation. In *EMNLP*, pages 4306–4315.
- Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and effective text matching with richer alignment features. In *Association for Computational Linguistics (ACL)*.
- Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.
- Tao Zhang, Jin Zhang, Chengfu Huo, and Weijun Ren. 2019. Automatic generation of pattern-controlled product description in e-commerce. In *WWW*, pages 2355–2365.

A Dataset

A.1 Data Collection

Since there are few previous studies about the post generation for advertisement, there is not any public dataset that we can use directly. Thus we construct an multi-product AD post dataset from scratch and our raw data is coming from a content scenario platform named Bimai Qingdan (which means a list of goods you must buy) in Taobao. Millions of posts are composed of professional paid copywriters to introduce and recommend different products for online shoppers to make the products more popular. Each post consists of several items (products) with their images, description copywriting, and a title that reflects the topic of this post. For example, a post that recommends some make-ups may consist of products like different perfumes and have a title like “start a new gorgeous day and become the most beautiful lady in the street.”

We extract the title of each post as the post topic T . To construct product candidate set P introduced in Section 3, we sample negative samples from the product set which have the same categories (such as food, clothes and so on.) with the positive samples, i.e., the products in the real post. After we select the products in the candidate set, we obtain the corresponding short product description title which contains the attribute keywords and regard them as product information $\{u_1^p, u_2^p, \dots, u_{n_p}^p\}$ mentioned in Section 3. Naturally, the long product description copywriting of all products in the post are regarded as the target C for our work. We keep all the posts that contain six products and the number of the negative samples is fixed to nine in all experiments.

A.2 Data Filtering

To clear our data, we filter the text length of each post and guarantee all post title consists of 7 to 14 words, the lengths of each product title are between 3 and 15, and the description text of each product has more than 10 words and less than 35 words. Finally, we obtain a clean and reasonable dataset through the above filtering operations. Finally, we obtain 339433 posts and randomly split them into 330,000/4,433/5,000 as training/validation/testing sets. More details are shown in Table 6.

Table 6: Some statistics about our dataset.

Text Type	Post Title	Product Title	Product Description
Min Length	7	3	10
Max Length	14	15	35
Size	339k	5,085k	2,034k

B More Implementation Details

B.1 Pretraining

To enhance the performance of our model, we conducted two pre-training procedures. First, we use each product information and its corresponding copywriting in the training data to build a pre-training dataset and use this pre-training dataset to pretrain the encoder-encoder framework, which consists of the product encoder in SelectNet and Decoder consist of the agents (RNN) in MGenNet. We save the parameters when the loss is the lowest on the eval dataset. Based on the parameters pretrained in the first step, SelectNet is pre-trained. The process keeps about 20 epoch and proves the SelectNet can converge.

B.2 Configurations

All our proposed models are implemented in PyTorch (Paszke et al., 2019) version 1.0 and Python 3.6. The experiments are conducted on a Linux server equipped with 8 NVIDIA V100-SXM2-16GB GPUs. The system of the Linux server is the AliOS.

B.3 Real-Life E-commerce Scenario

Product candidate sets in the real-life e-commerce scenario in Taobao Mobile are always too huge to be used directly into our model. Targeted to this issue, we use the retrieval tool ElasticSearch⁶ to search the products which are related to the key words we want to express in the multi-product AD post title. We rank those products by the related degree and obtain the Top-15 or Top-20 related products as the new product candidate set.

⁶<https://www.elastic.co/>