Poet: Product-oriented Video Captioner for E-commerce

Shengyu Zhang, Ziqi Tan, Jin Yu, Zhou Zhao, Kun Kuang, Jie Liu, Jingren Zhou

ACM Reference Format:

1 INTRODUCTION

Nowadays, a growing number of short videos are generated and uploaded to Taobao. Among these videos, user-generated videos are massive in volume and share unique product experiences, such as the individual preference for the product usage scenario or usage strategy. When recommending these videos to customers for product promotion, accompanying a description that narrates the uploader-preferred highlights depicted in the product video is essential for successful promotion, i.e., attracting potential customers with similar interests or preferences to buy the same product. Different from traditional video-to-text generation problems which mainly concern what exists or happens in the video, this problem cares about what the video uploader wants to highlight. We name this particular problem as product-oriented video captioning.

Product-oriented video captioning naturally requires a fine-grained analysis of prominent product characteristics depicted in the video. However, without some general understanding of the product, it can be hard even for a human to grasp what the uploader mainly concerns based on the isolated video. To this end, we view leveraging product-related knowledge as a fundamental ability for product-oriented video captioning. Concretely, we take the structured product aspects from the associated product as prior knowledge since they are easy to acquire (most user-generated videos in e-commerce have on-sell product associations.) and concise in meaning. The structured product aspects arranged by product-sellers contain general and basic information necessary for fine-grained video understanding. Figure 1 reveals the task definition, the application scenario in Mobile Taobao, and how automatic tools contribute to product promotion.

Recent advances in deep neural networks [18, 33, 41], especially the RNNs, have convincingly demonstrated high capability in handling the general video captioning problem. Most of them [22, 37, 51] incorporate an RNN based encoder-decoder structure with/without attention mechanisms to perform sequential frames encoding and sequential words decoding. However, product-oriented video captioning poses some unique challenges. A first
To summarize, this paper makes the following key contributions:

- We propose to investigate a real-world video-to-text generation problem, product-oriented video captioning, to automatically narrate the user-preferred product characteristics inside user-generated videos.
- We propose a novel Poet framework for product-oriented video captioning via simultaneously capturing the user-preferred product characteristics and modeling the dynamic interactions between them with a product-oriented spatial-temporal graph.
- We introduce a novel knowledge leveraging module to incorporate the product aspects for product video analysis by performing knowledge filtering, dynamic memory writing, and knowledge attending. Poet yields consistent qualitative and qualitative improvements on two product-oriented video captioning datasets 1 that are collected in Mobile Taobao.

2 RELATED WORKS

2.1 Video to Text Generation


More recently, there are works exploiting object-level features in representing the videos [11, 40, 49, 54] for video description generation. They mainly propose to detect salient objects and employ RNNs to model the temporal structure between them. However, they are not directly applicable to product-oriented video captioning for the following two reasons: 1) product-oriented video captioning requires even more fine-grained video analysis, i.e., product-part characteristic recognition. 2) These methods neglect the spatial interactions between region-region and region-background within frames. Poet represents both the detected product-parts and the whole frames as spatial-temporal graphs and employs the graph neural network to model the interactions between product-parts and product-background.

2.2 Knowledge enhanced Video Analysis

Incorporating external in-domain knowledge is a promising research direction [23] for video analysis. There are mainly two kinds of external knowledge, i.e., knowledge graph and topically related documents (e.g., Wikipedia). Knowledge graph based methods [6, 7] typically retrieve the knowledge graph from off-the-shelf knowledge bases such as ConceptNet 5.5 [31] and employ the graph convolution network [12] to perform knowledge reasoning. These methods are not suitable for our task since there are no well-defined relationships among product aspects. For document-based approaches, Venugopal et al. [36] uses the Wikipedia corpus to
pre-train a language model (LM) and proposes the late/deep fusion strategies to enhance the decoding RNN with the LM. Whitehead et al. [46] first retrieves the relevant document and then use the pointer mechanism to directly borrow entities in the decoding stage. Different from these works, Poet performs knowledge leveraging in the product-oriented spatial-temporal inference stage.

3 METHODS

3.1 Overview

After data preprocessing (details in Section 4.1), we represent each product-oriented video as frame-level features \( \{ f_i \in \mathbb{R}^{D_f} \}_{i=1, \ldots, N_f} \) where \( f_i \) is the \( D_f \) length feature vector for the \( i \)th frame \( f_i \), and product-part features \( \{ p_{i,j} \in \mathbb{R}^{D_p} \}_{j=1, \ldots, N_p} \) where \( p_{i,j} \) is the \( D_p \) length feature vector for the \( j \)th product-part in the \( i \)th frame. The video-associated product aspects are \( \{ a_k \}_{k=1, \ldots, N_a} \) and we use the embedding layer to learn an aspect embedding \( a_k \in \mathbb{R}^{D_a} \) of dimension \( D_a \) for the \( k \)th aspect \( a_k \). We aim to generate a video description \( \{ w_m \}_{m=1, \ldots, N_w} \) that narrates the preferred product characteristics of e-commerce buyers/fans.

We firstly build a product-oriented spatial-temporal video graph (See Figure 2), which contains both frame nodes and product-part nodes. With the graph representation, the encoder of Poet mainly incorporates two sub-modules, i.e., the spatial-temporal inference module for graph modeling, and the knowledge leveraging module for product aspects modeling. These sub-modules can be easily stacked to obtain a progressive knowledge retrieval and knowledge enhanced visual inference process. In the next several subsections, we will formally introduce the building blocks comprising Poet, including the graph building process, the spatial-temporal inference module, the knowledge leveraging module, and the attentional RNN-based decoder in detail.

3.2 Product-oriented Video Graph Modeling

3.2.1 Graph Building. To better capture the highlights (i.e., preferred product characteristics) inside the product videos, we propose to represent the videos as spatial-temporal graphs.

Nodes Different from previous works [11, 40, 49, 54] that represent the objects as graph nodes, we represent product parts as nodes to capture the dynamic change of these fine-grained details along the timeline. Product-part features are extracted by a pre-trained CNN-based detector (details in 4.1), and thus these features naturally contain spatial cues. However, since we do not model the product parts along the timeline using RNNs, there is no concept of frame order in the modeling process. To this end, we add an order-aware embedding \( o_i \in \mathbb{R}^{D_p} \) to each product-part feature, which is similar to the position embedding strategy employed in sequence learning [8, 20]. \( o_i \) stands for the embedding for the frame order \( i \). Sharing similar spirit, we further obtain the type-aware product-part representation by adding the part-type embedding \( t_j \in \mathbb{R}^{D_p} \), which stands for the \( j \)th part for a particular product, such as waistline and hem. Then, the enhanced product-part feature \( p_{i,j}^e \) can be obtained by:

\[
p_{i,j}^e = p_{i,j} + o_i + t_j.
\]

Besides the product-part nodes, we further incorporate the frame node into each frame graph to capture the product as a whole and exploit the correlations between the products and the backgrounds. Similar to the product-part feature, we add the order-based embedding \( o_i \) and a special type embedding \( t_{\text{frame}} \) for obtaining the frame-order concept and the frame-type concept, respectively.

\[
f_i^e = f_i + o_i + t_{\text{frame}}.
\]

We then project the product-part features and the frame features into a common space by employing two linear transformations, i.e., \( W_p \in \mathbb{R}^{D_p \times D_p} \) and \( W_f \in \mathbb{R}^{D_p \times D_f} \).

\[
\begin{align*}
v_{i,j}^p &= W_p p_{i,j}^e + b_p, \\
v_{i,\text{frame}} &= W_f f_i^e + b_f.
\end{align*}
\]

Edges To capture the correlations among product-parts within the same frame as well as the interactions between global frame context (e.g., background, the product as a whole) and local part details, we propose a baseline method by fully connecting the product-part nodes and frame node within each frame graph. To obtain a comprehensive understanding and the dynamic change of product-parts across different viewpoints, we fully connect the nodes of the same type (including the frame type) from all frames. The edge weights are obtained using a fully-connected layer:

\[
e_{i,k} = W_e [v_i, v_k] + b_e,
\]

where \( e_{i,k} \in \mathbb{R} \) is the weight of edge between node \( v_i \) and \( v_k \). We use a linear transformation \( W_e \in \mathbb{R}^{1 \times 2D_p} \) with a bias term \( b_e \in \mathbb{R} \) to estimate the correlation of two nodes. \([\cdot,\cdot]\) denotes the concatenation operation. For convenience, we use \( G_e \) to denote the initial video graph and intermediate video graphs since only nodes feature representations are updated.

3.2.2 Spatial-temporal Inference. Although previous works [11, 40, 49] have proposed to capture the fine-grained region-of-interests such as objects and [54] proposes to represent these fine-grained cues as graphs, they all use RNN-based modeling which can be inefficient for its internally recurrent nature and can be less effective in modeling interactions of regions within a frame since these regions have no natural temporal dependencies. To this end, we employ the flexible graph neural networks for spatial-temporal inference. Existing works performing video graph modeling for video relation detection [25], temporal action localization [53], and video action classification [43] often leverage the off-the-shelf Graph Convolutional Networks [13] for information propagation. We propose a new modeling schema by separately modeling the root node and neighbor nodes when aggregation. For neighbor nodes information aggregation, we use an element-wise max function for its effectiveness in the experiment:

\[
\bar{v}_{i,c} = \max_{k \in \mathcal{N}(v_i)} \{ e_{i,k} \ast v_{k,c} \},
\]

where \( \mathcal{N}(v_i) \) denotes the neighbor nodes set of the root node \( v_i \) and \( v_{k,c} \) is the \( c \)th element in the feature vector of node \( v_k \). We note the edge weight \( e_{i,k} \) will be re-computed for each information
propagation process. We then perform separate modeling for the root node and the neighbor nodes:

\[
\hat{v}_i = W_n \hat{v}_i^n + b_n + W_r r_i + b_r,
\]

(6)

where \(W_n, W_r\) are linear transformations to project the root representation and the aggregated neighbors representation into a common space. \(b_n, b_r\) are the bias terms. This schema further incorporates an element-wise function for re-weighting the importance of each position as well as a short-cut connection:

\[
\hat{v}_i = \sigma(W_n^a \hat{v}_i^n + b_n^a + W_r^a r_i + b_r^a) \cdot \hat{v}_i + v_i.
\]

(7)

where \(\cdot\) denotes the Hadamard product. Matrices \(W_n^a, W_r^a\) and the corresponding biases \(b_n^a, b_r^a\) model the position-wise importance. \(\sigma\) denotes the element-wise sigmoid function.

3.3 Product-aspects Leveraging

3.3.1 Knowledge Filtering. Leveraging product aspects as knowledge is an essential part of obtaining the basic product information first and a better understanding of the user-generated product videos. Different from other kinds of external knowledge (e.g., Knowledge Base and Wikipedia), the aspects of the associated product contain noised values that may hurt the performance of product video understanding. For example, there can be both black-white and red-blue color choices for a certain t-shirt on sell while the buyer/fan may love the black-white one and wear it in the video. We therefore devise a knowledge filtering module based on the hard attention mechanism to filter noised values such as red-blue for each video. Formally, given the nodes features \(\{v_r\}_{r=1,\ldots,N_F+N_p+1}\) in the video graph \(G_v\) and product aspect embeddings \(\{a_k\}_{k=1,\ldots,N_a}\), we perform knowledge filtering by:

\[
v^o = \frac{1}{N_F \ast (N_p + 1)} \sum_r v_r,
\]

(8)

\[
\alpha_k = \sigma(W_h[a_k, v^o] + b_h),
\]

(9)

\[
\hat{A} = \{a_k \cdot \exp(\alpha_k) > \beta_k\},
\]

(10)

where \(\hat{A}\) denotes the filtered aspect set, which includes aspect \(a_k\) with importance \(\alpha_k\), over a certain threshold \(\beta_k\). We empirically set the threshold to the uniform probability \(1/N_a\). The importance indicator \(\alpha_k\) is computed using a linear transformation \(W_h\) and a bias term \(b_h\). We use the global (or averaged-pooled) representation \(v^o\) of the video graph as the filtering context since we aim to remove aspects that are irrelevant to any part of the video. We add a sigmoid function \(\sigma\) to prevent large importance scores, which may lead to a small filtered aspect set after the scores being forward to the \(\text{softmax}\) function.

3.3.2 Dynamic Memory Modeling. Previous works that incorporate external knowledge for video description generation [36, 46] often leverage the knowledge in the decoding stage, i.e., using pointer mechanism to directly borrow the words/entities from the knowledge document or using attention mechanism to update the decoder hidden state. We propose to progressively retrieve relevant knowledge in the encoding stage, which enables a better understanding of the video for capturing user-preferred product highlights. Specifically, we employ a memory network [9, 32, 45] based approach and enhance it with dynamic memory writing:

\[
a_s = \alpha_s \ast (W_o a_s + b_s) + (1 - \alpha_s) \ast (W_g v^o + b_g),
\]

(11)

\[
\omega_{r,s} = W_o \tanh(W_m [v_r, \tilde{a}_s] + b_m),
\]

(12)

\[
\hat{v}_r = g \ast v_r + \sum_s \omega_{r,s} \ast \tilde{a}_s, \text{ where } \omega_{r,s} = \frac{\exp(\omega_{r,s})}{\sum_o \exp(\omega_{r,o})}
\]

(13)
where the memory writing process (Equation 11) borrows the importance factor $a_t$ from Equation 9 (Note that the importance factor $a_t$ is in the range $(0, 1)$ after the $σ$ function.) This process helps inhibit irrelevant aspect information (with smaller $a_t$) and enliven the more relevant ones (with larger $a_t$). $γ$ controls to what extent the final representation $\hat{v}_r$ depends on the initial representation $v_r$ and we empirically set it to 0.5.

### 3.4 Progressive Inference Encoder

Since the spatial-temporal inference module and the knowledge leveraging module updates the node representation without modifying the graph structure, we can easily stack multiple STI modules and multiple KL modules. Poet builds the inference encoder by progressively and alternatively performing STI and KL as depicted in Figure 2. In such a design, we aim to not only obtain higher-order graph reasoning (i.e., with access to remote neighbors) but also propagate the leveraged knowledge to the whole graph by the following STI modules. We denote the combination of one STI and one KL as one graph reasoning layer. We use two-layers graph reasoning in the experiment and we observe stacking more layers, which may make node representations over-smoothing and not distinct (i.e., all nodes contain similar information), will lead to a minor performance drop.

### 3.5 Decoder

Following many previous works [18, 57], we build the decoder with the RNN (here we use gated recurrent unit GRU [5]) and soft attention mechanism. We first initialize the hidden state of GRU as the global representation of the knowledge-aware video graph:

$$h_0 = v^\top = \frac{1}{N_f \times (N_p + 1)} \sum_{t} v_r,$$

(14)

For each decoding step $t$, we attend to each node inside the video graph and aggregate the visual cues using the weighted sum:

$$g_t = W_{h} \tanh(W_{md}[v_t, h_t] + b_{md}),$$

(15)

$$h_t = \sum_{t} g_t \times v_r,$$

(16)

where $W_{h}, W_{md}$ are linear transformations and they together model the additive attention [5] with the bias term $b_{md}$. $\tilde{g}$ is the attention weights. The $t$th decoding process can be formulated as:

$$h_{t+1} = GRU(h_t, [\tilde{w}_t, \tilde{h}_t]),$$

(17)

where $\tilde{w}_t$ denotes the embedding of the predicted word $w_t$ at step $t$. For training objectives, we take the standard cross-entropy loss:

$$\mathcal{L} = - \sum_{t} \log p(w_t).$$

(18)

### 4 EXPERIMENTS

#### 4.1 Product-oriented Video Datasets

**Data Collection** We collect two large-scale product-oriented video datasets, i.e., buyer-generated fashion video dataset (BFVD) and fan-generated fashion video dataset (FFVD) for *product-oriented* video captioning research. Data samples from both datasets are collected from Mobile Taobao. We collect the videos, the descriptions, and the associated product aspects to form the datasets. Each recommended video has been labeled as buyer-generated or fan-generated by the platform. These two kinds of data are originally generated by users with different background knowledge and intentions. Buyers often focus on the general appearance, salient characteristics, and emotions while descriptions generated by fans often reflect deep insights and understandings about the products. Therefore, we regard these two kinds of videos as individual datasets since they may pose different challenges for modeling. Data collected from real-world scenario may contain noises and we select videos with PV (page views) over 100,000 and with CTR (click through rate) larger than 5%. Videos and descriptions of high quality are more likely to be recommended (more PVs) and clicked (bigger CTR).

**Data Statistics** As a result, we collect 43,166 *video, aspects, description* triplets in BFVD and 32,763 triplets in FFVD. The basic statistics and comparison results with other frequently used video captioning datasets are listed in Table 1. The distinguishing characteristics of BFVD and FFVD are 1) these datasets can be viewed as an early attempt to promote video-to-text generation for the domain of e-commerce. 2) Concerning the total number and total length of videos, BFVD and FFVD are among the largest. 3) As for language data, BFVD and FFVD contain a large number of unique words, and the ratio of $\frac{\#\text{Vocab}}{\#\text{Sentences}}$ is among the largest. These statistics indicate that BFVD and FFVD contain abundant vocabulary and little repetitive information. 4) BFVD and FFVD associate corresponding product aspects, which permit not only the product-oriented video captioning research as we do but also a broad range of product-related research topics such as multi-label video classification for e-commerce.

**Data Preprocessing** For descriptions, we remove the stop words and use Jieba Chinese Tokenizer \(^2\) for tokenization. To filter out those noised expressions such as brand terms and internet slang terms, we remove tokens with frequency less than 30. Descriptions with tokens more than 30 will be shortened, and the max length for product aspects is 12. Following the standard process, we add a `<sos>` at the beginning of each description and a `<eos>` at the last.

---

2https://github.com/fxsjy/jieba

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>#Videos</th>
<th>#Sentence</th>
<th>#Vocab</th>
<th>Dur(hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSVD [3]</td>
<td>open</td>
<td>1,970</td>
<td>70,028</td>
<td>13,010</td>
<td>5.3</td>
</tr>
<tr>
<td>TACos [26]</td>
<td>cooking</td>
<td>123</td>
<td>18,227</td>
<td>28,292</td>
<td>15.9</td>
</tr>
<tr>
<td>TACos M-L [27]</td>
<td>cooking</td>
<td>185</td>
<td>14,105</td>
<td>-</td>
<td>27.1</td>
</tr>
<tr>
<td>MPii-MD [28]</td>
<td>movie</td>
<td>94</td>
<td>68,375</td>
<td>24,549</td>
<td>73.6</td>
</tr>
<tr>
<td>M-VAD [34]</td>
<td>movie</td>
<td>92</td>
<td>55,905</td>
<td>18,269</td>
<td>84.6</td>
</tr>
<tr>
<td>VTW [52]</td>
<td>open</td>
<td>18,100</td>
<td>-</td>
<td>23,059</td>
<td>213.2</td>
</tr>
<tr>
<td>MSR-VTT [47]</td>
<td>movie</td>
<td>7,180</td>
<td>200,000</td>
<td>29,316</td>
<td>41.2</td>
</tr>
<tr>
<td>Charades [30]</td>
<td>human</td>
<td>9,848</td>
<td>-</td>
<td>-</td>
<td>82.01</td>
</tr>
<tr>
<td>ActivityNet [14]</td>
<td>activity</td>
<td>20,000</td>
<td>10,000</td>
<td>-</td>
<td>849</td>
</tr>
<tr>
<td>DiDeMo [10]</td>
<td>open</td>
<td>10,464</td>
<td>40,543</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>YouCook2 [55]</td>
<td>cooking</td>
<td>2,000</td>
<td>-</td>
<td>2,600</td>
<td>175.6</td>
</tr>
<tr>
<td>VATEX [44]</td>
<td>open</td>
<td>41,300</td>
<td>826,000</td>
<td>82,654</td>
<td>-</td>
</tr>
</tbody>
</table>

---

Table 1: Comparing BFVD and FFVD with exiting video-to-text datasets (e-comm stands for e-commerce).
For videos, we inspect 30 frames for each video and extract the product-part features for each frame using this detector [17] pretrained on this dataset [19]. For each frame, the product-part detector will produce a prediction map with the same width/height as the input frame. Each pixel value at a particular part-channel (note that the number of channels is the same as the number of product parts) indicates how likely this pixel is belonging to this part. We then apply the prediction map on the activations obtained from an intermediate layer (by resizing the probability map first, apply softmax on this map to obtain probability weights and finally weighted sum over the intermediate activations per part channel) to obtain part features. Specifically, we use the activations from pooled_5 and result in $8 \times 64$ features vectors for $8$ product-parts. We mean-pool the activations from layer conv4 as the representation for each frame.

For train/val/test split, we employ a random sampling strategy and adopt 65%/5%/30% for the training set, validation set, and testing set, respectively. Therefore, we have 21,554/1,658/9,948 data samples for FFVD and 28,058/2,158/12,950 data samples for BFVD.

### 4.2 Evaluation Measurements

**Natural Language Generation Metrics** Concretely, we adopt four numerical assessment approaches, BLEU-1 [24] for sanity check, METEOR [1] based on unigram precision and recall, ROUGE-L [16] based on the longest common subsequence co-occurrence, CIDEr [35] based on human-like consensus. With the user-written sentences as references, these measurements can help evaluate the generation fluency as well as whether the generation model captures the user-preferred highlights depicted in the video.

**Product Aspects Prediction** Since leveraging product aspects as knowledge is essential for product video understanding, it is necessary to evaluate how well the generation model captures such information. We follow the evaluation protocol proposed in KOBE [4] and view the aspects prediction accuracy as the indicator.

**Lexical Diversity** The above measurements mainly concern the generation quality (fluency, highlight capturing, and aspects capturing), and they cannot explicitly evaluate the generation diversity. As we know, repetitive phrases and general descriptions can be less attractive for potential buyers. Therefore, as in KOBE [4], we view the number of unique n-grams of the sentences generated when testing as the indicator of the generation diversity. We empirically choose 4-grams and 5-grams, following KOBE.

### 4.3 Comparison Baselines

To evaluate the effectiveness of Poet, we compare it with various video description baselines. Since most baselines concern only video information, we re-implement these baselines and add separate encoders (with a similar structure to their video encoders) for product aspect modeling.

1. **AA-MPLSTM.** Aspect-aware MPLSTM (originally [38]) employs two mean-pooling encoders and concatenate the encoded vectors as the decoder input.
2. **AA-Seq2Seq.** Aspect-aware Seq2Seq (originally [37]) uses an additional RNN encoder for aspect modeling.
3. **AA-SALSTM.** Aspect-aware SALSTM (originally [51]) equips the RNN decoder in AA-S2VT with soft attention.
4. **AA-HRNE.** Aspect-aware HRNE (originally [22]) differs the AA-S2VT by employing hierarchical encoders.
5. **AA-RecNet.** For Aspect-aware RecNet, we re-construct not only the frame features (as in RecNet [39]) but also the aspect features. Also, there is an additional aspect encoder.
6. **Unified-Transformer.** We modify the Unified Transformer [21] by replacing the live comments encoder with an aspect encoder.

7. **PointerNet.** We equip the Seq2Seq model with the entity pointer network, proposed by Whitehead et al.[46], for product aspect modeling.

### 4.4 Performance Analysis

In this section, we examine the empirical performance of Poet on two product-oriented video datasets, i.e., BFVD and FFVD. We concern a couple of the following perspectives.

**Overall generation quality.** By generation quality, we mean both the generation fluency and whether the generated sentences capture user-preferred product characteristics. Referenced-based natural language metrics can help reflect the performance since they directly compare the generated sentences to what the video uploaders describe. In a nutshell, the clear improvement over various competitors and across four different metrics demonstrate the superiority of the proposed Poet (as shown in Table 2, NLG metrics). Specifically, on BFVD, we obtain $+2.46$ BLEU (relatively 20.3%) and $+0.77$ METEOR (relatively 12.1%) improvement over the best (PointerNet) among the competitors. For FFVD, we observe that fan-generated videos sometimes concern collections of clothes and present them for a specific theme (such as dressing guide or clothes that make you look fit) while only one of the clothes (in the same video) is associated with product aspects. Such a phenomenon may introduce noise and hurt the performance of models designed for single-product modeling. Nevertheless, Poet achieves the best performance across the most metrics. We attribute the clear advantage over other designs to 1) **product-oriented video graph modeling.** Poet represents product parts across frames as spatial-temporal graphs, which can better capture the dynamic change of these characteristics along the timeline and find out those distinguishing highlights that are preferred by the user (e.g., distinguishing characteristics of one product part can be highlighted in higher frequencies or with close-up views). In contrast, previous models (including the RNN-based and the transformer-based) are inferior in capturing such characteristics since they either model only the frame-level features without fine-grained analysis or only model the videos in a sequential way. 2) **Knowledge enhanced video analysis.** Poet firstly perform hard attention to remove noise aspects that are of no use for video analysis and then perform dynamic memory writing/attending to progressively enhance the spatial-temporal inference process. This design can be superior over other designs like concatenation or the complex PointerNet design. We will further analyze different knowledge incorporation methods in **Aspect Capturing** and **Ablation studies.**

**Aspect Capturing.** Knowledge incorporation methods can be categorized into three sub-groups: 1) **AA- methods.** We transform
a video captioning method into a AA-method by adding a separate encoder, which has a similar structure to the existing visual encoder (such as the hierarchical RNN encoder in HRNE), for the product aspect modeling. We concatenate the encoded features of the video and the encoded visual feature as the initial decoder input.

2) Decoding-oriented methods. Unified-transformer and PointerNet are decoding-oriented knowledge incorporation methods, which introduce the external knowledge in the decoding stage as an implicit or explicit reference. Unified-transformer implicitly utilizes the knowledge using the multi-head attention mechanism before prediction. PointerNet explicitly viewed the attended entities (product aspects) as candidate words for prediction and combine the attention weights and the vocabulary probability distribution before prediction. 3) Poet: the analysis-oriented method which aims to obtain a better understanding of the videos with the external knowledge as guidance. The aspect capturing scores are shown in Table 2 (Aspect Prediction). It can be seen that 1) the analysis based method (Poet) achieves the best performance on two datasets. As we illustrated earlier, the product-oriented video captioning requires a fine-grained analysis of distinguishing characteristics depicted in the video, and the product aspects can better serve as the prior background knowledge to obtain such kind of analysis. 2) The decoding-oriented methods cannot beat the AA-methods on BFVD. This is a reasonable result since the product aspect can be associated with only one cloth in the fan-generated video (as we stated in the previous section). The PointerNet achieves the best concerning 4-grams in BFVD, which is reasonable since they externally consider the aspects as candidate decoding words.

Table 2: Qualitative results of the proposed Poet with diverse competitors. Comparisons concern generation quality (NLG metrics), product aspect capturing, and generation diversity. Poet achieves the best results on two product-oriented video captioning datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Methods</th>
<th>BLEU-1</th>
<th>METEOR</th>
<th>ROUGE_L</th>
<th>CIDEr</th>
<th>Aspect Prediction</th>
<th>Lexical Diversity</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFVD</td>
<td>AA-MPLSTM</td>
<td>11.31</td>
<td>6.02</td>
<td>10.08</td>
<td>9.76</td>
<td>54.31</td>
<td>2.94</td>
</tr>
<tr>
<td></td>
<td>AA-Seq2Seq</td>
<td>11.96</td>
<td>6.14</td>
<td>11.05</td>
<td>11.67</td>
<td>54.85</td>
<td>4.74</td>
</tr>
<tr>
<td></td>
<td>AA-HRNE</td>
<td>11.82</td>
<td>5.98</td>
<td>10.23</td>
<td>11.86</td>
<td>55.98</td>
<td>5.02</td>
</tr>
<tr>
<td></td>
<td>AA-SALSTM</td>
<td>11.78</td>
<td>5.88</td>
<td>10.18</td>
<td>11.57</td>
<td>55.93</td>
<td>5.10</td>
</tr>
<tr>
<td></td>
<td>AA-RecNet</td>
<td>11.17</td>
<td>6.01</td>
<td>11.05</td>
<td>11.67</td>
<td>54.94</td>
<td>5.06</td>
</tr>
<tr>
<td></td>
<td>Unified-Transformer</td>
<td>11.28</td>
<td>6.32</td>
<td>10.43</td>
<td>12.66</td>
<td>55.12</td>
<td>3.35</td>
</tr>
<tr>
<td></td>
<td>PointerNet</td>
<td>12.09</td>
<td>6.34</td>
<td>11.19</td>
<td>12.58</td>
<td>56.01</td>
<td>5.36</td>
</tr>
<tr>
<td>FFVD</td>
<td>AA-MPLSTM</td>
<td>14.55</td>
<td>7.11</td>
<td>12.13</td>
<td>13.48</td>
<td>56.69</td>
<td>5.16</td>
</tr>
<tr>
<td></td>
<td>AA-Seq2Seq</td>
<td>14.77</td>
<td>7.87</td>
<td>13.74</td>
<td>18.54</td>
<td>62.01</td>
<td>4.08</td>
</tr>
<tr>
<td></td>
<td>AA-HRNE</td>
<td>13.58</td>
<td>6.75</td>
<td>12.06</td>
<td>20.10</td>
<td>60.39</td>
<td>4.32</td>
</tr>
<tr>
<td></td>
<td>AA-RecNet</td>
<td>15.11</td>
<td>8.03</td>
<td>14.18</td>
<td>19.08</td>
<td>62.21</td>
<td>4.45</td>
</tr>
<tr>
<td></td>
<td>Unified-Transformer</td>
<td>14.39</td>
<td>7.42</td>
<td>13.45</td>
<td>21.00</td>
<td>62.01</td>
<td>3.39</td>
</tr>
<tr>
<td></td>
<td>PointerNet</td>
<td>15.28</td>
<td>7.77</td>
<td>14.02</td>
<td>18.85</td>
<td>61.30</td>
<td>4.40</td>
</tr>
</tbody>
</table>

| Poet    | 16.04  | 8.06   | 14.82   | 21.71  | 62.70           | 4.60             |

<table>
<thead>
<tr>
<th>Models</th>
<th>Fluency</th>
<th>Diversity</th>
<th>Overall Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA-RecNet</td>
<td>2.73</td>
<td>3.49</td>
<td>3.04</td>
</tr>
<tr>
<td>AA-Transformer</td>
<td>2.66</td>
<td>3.37</td>
<td>2.95</td>
</tr>
<tr>
<td>Poet</td>
<td>2.88</td>
<td>3.59</td>
<td>3.15</td>
</tr>
</tbody>
</table>

Human Evaluation We agree with AREL [42] that human judgement is essential for stable evaluation especially when the captions are highly diverse and creative. Following the human evaluation protocol of KOBÉ [4] and Li et al. [15], we randomly select 1,000 instances from the testing set and distribute them to human annotators. The results are listed in Table 3. Compared to the typical Transformer-based design and RNN-based design (AA-RecNet), Poet generates more fluent (fluency +0.22/+0.15) and diversified (diversity +0.22/+0.10) descriptions. The indicator Overall Quality reflects whether the descriptions capture the product characteristics and video uploader highlights in the video. In terms of this metric, Poet still demonstrates a clear performance improvement over AA-Transformer/AA-RecNet by +0.20/+0.11.

Case Study Figure 3 shows two generation cases on the FFVD testing set. In summary, Poet generates more fluent and complete sentences than the AA-Transformer and AA-Recnet, which are typical architectures of transformer-based and RNN-based models, respectively. For example, the phrases (such as "to in") generated by AA-Transformer in the first case are confusing, and the whole sentence is incomplete. Besides the generation fluency, Poet can generate sentences that better capture the product aspects. In the second case, the phrases "soft" and "young fashionistas" are derived from the aspects "soft elastic", "youth", and "fashion".
Figure 3: Generation samples of Poet, AA-Transformer and AA-Recnet on the FFVD testing set. We present the filtered aspects and the corresponding scores in the proposed knowledge leveraging module.

Table 4: Ablation study on the generation quality of Knowledge Leveraging module and the pointer mechanism.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Methods</th>
<th>BLEU-1</th>
<th>METEOR</th>
<th>ROUGE_L</th>
<th>CIDEr</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFVD</td>
<td>Poet</td>
<td>14.55</td>
<td>7.11</td>
<td>12.13</td>
<td>13.48</td>
</tr>
<tr>
<td></td>
<td>+ pointer</td>
<td>13.26</td>
<td>6.60</td>
<td>11.53</td>
<td>13.18</td>
</tr>
<tr>
<td></td>
<td>- KL</td>
<td>12.43</td>
<td>6.48</td>
<td>10.86</td>
<td>12.25</td>
</tr>
<tr>
<td>FFVD</td>
<td>Poet</td>
<td>16.04</td>
<td>8.06</td>
<td>14.82</td>
<td>21.71</td>
</tr>
<tr>
<td></td>
<td>+ pointer</td>
<td>16.13</td>
<td>7.79</td>
<td>14.50</td>
<td>20.57</td>
</tr>
<tr>
<td></td>
<td>- KL</td>
<td>15.53</td>
<td>7.89</td>
<td>14.18</td>
<td>19.73</td>
</tr>
</tbody>
</table>

To further demonstrate the effectiveness of the proposed knowledge leveraging module, we extract and present the filtered aspects with the corresponding scores. We observe that the proposed KL module successfully filters those aspects of no use, such as product sizes (XL) and the release year 2019 in the first case, and the scores of remaining aspects are consistent to the contribution to the final description. Also, Poet can generate creative while accurate words (e.g., casual) beyond the input aspects set based on its understanding of the video.

4.5 Ablation Studies

We conduct ablation studies to verify the effectiveness of proposed modules within Poet. We mainly concern the following two issues:

- When modeling the video as a graph, does the knowledge leveraging module outperform the pointer mechanism? To answer this question, we construct two models, i.e., the Poet+pointer model, which directly adds the pointer mechanism to Poet, and the Poet+pointer-KL model, which removes the proposed knowledge leveraging module from Poet+pointer model. The experiment results on two datasets are listed in Table 4. It can be seen that 1) adding pointer mechanism may hurt the performance in most cases. This further demonstrates the superiority of the analysis-oriented knowledge incorporation method for product-oriented video captioning. 2) removing the knowledge leveraging module leads to a clear performance drop (-0.93 CIDEr and relatively 7% in BFVD), which shows the effectiveness of the proposed module.

- Do all the proposed modules improve the generation quality? We surgically remove the proposed modules and individually test the performance on BFVD. Table 5 shows the numeric results.

We note that removing the KL (knowledge leveraging) module means ignoring the external product aspects totally. The result indicates the merit of leveraging the KL module to enhance the fine-grained video analysis. By “SPI”, we replace the SPI (spatial-temporal inference) module by the popular Graph Convolutional Networks [12]. The improvement over the GCN verifies the effectiveness of the spatial-temporal inference module.

5 CONCLUSION

In this paper, we propose to narrate the user-preferred product characteristics depicted in user-generated product videos, in natural language. Automating the video description generation process helps video recommendation systems in e-commerce to leverage the massive user-generated videos for product promotion. We propose a novel framework named Poet to perform knowledge-enhanced spatial-temporal inference on product-oriented video graphs. We conduct extensive experiments including qualitative analysis, ablation studies, and numerical measurements concerning generation quality/diversity. Experiment results show the merit of video graph modeling, the proposed spatial-temporal inference module, and the knowledge leveraging module for the product-oriented video captioning problem. We collect two user-generated fashion video datasets associated with product aspects to promote not only the product-oriented video captioning research, but also various product-oriented research topics such as product video tagging.

6 ACKNOWLEDGMENTS

The work is supported by the NSFC (61625107, 61751209, 61836002), National Key R&D Program of China (No. 2018AAA0101900, No. 2018AAA0100603), Zhejiang Natural Science Foundation (LR19F020006), Fundamental Research Funds for the Central Universities (2020QNA5024), and a research fund supported by Alibaba.


