

Received February 12, 2022, accepted March 2, 2022, date of publication March 8, 2022, date of current version May 5, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3157287

# Template-Based Headline Generator for Multiple Documents

YUN-CHIEN TSENG<sup>1</sup>, MU-HUA YANG<sup>1</sup>, YAO-CHUNG FAN<sup>2</sup>,  
WEN-CHIH PENG<sup>1</sup>, (Member, IEEE), AND CHIH-CHIEH HUNG<sup>3</sup>, (Member, IEEE)

<sup>1</sup>Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchiu 30010, Taiwan

<sup>2</sup>Department of Computer Science and Engineering, National Chung Hsing University, Taichung 40227, Taiwan

<sup>3</sup>Department of Management Information Systems, National Chung Hsing University, Taichung 40227, Taiwan

Corresponding author: Wen-Chih Peng (wcpeng@cs.nctu.edu.tw)


This work was supported by the Taiwan Ministry of Science and Technology under Grant MOST 110-2221-E-A49-164 and Grant MOST 110-2221-E-005-045.

**ABSTRACT** In this paper, we develop a neural multi-document summarization model, named MuD2H (refers to **Multi-Document to Headline**) to generate an attractive and customized headline from a set of product descriptions. To the best of our knowledge, no one has used a technique for multi-document summarization to generate headlines in the past. Therefore, multi-document headline generation can be considered new problem setting. Our model implements a two-stage architecture, including an extractive stage and an abstractive stage. The extractive stage is a graph-based model that identified salient sentences, whereas the abstractive stage uses existing summaries as soft templates to guild the seq2seq model. A series of experiments are conducted by using KKday dataset. Experimental results show that the proposed method outperforms the others in terms of quantitative and qualitative aspects.

**INDEX TERMS** Deep learning, graph convolutional network, headline generation, multiple documents summarization, natural language processing.

## I. INTRODUCTION

In the era of information explosion, people eager to find a way to acquire knowledge efficiently. To quick capture the ideas behind articles, people are likely to go through headlines first and then decide if an article is worthy to read. In this paper, we aims at generating headlines for multiple documents. Generating headlines for texts can be considered as a subproblem of summarization [1], [2]: The given sentence have to be representative and attractive. TemPEST [3] is a model design for generating personalized headlines which try to catch electronic direct mail receiver's attention. TemPEST is a soft template-based seq2seq model [4], including three stages: Retrieve, Rerank and Rewrite. The model builds an Information Retrieval (IR) system for index and search, reranks the search results then selects a suitable template. Beside summarize and abbreviate input document, a title generating model need to generate a suitable output. However, this work is design for single document. When we want to generate a representing sentence for a set of document, current model leads to a failure.

The associate editor coordinating the review of this manuscript and approving it for publication was Hao Luo .

To solve the problem, we proposed a model “from **Multi-Document to Headline**”, which generates a personalized headline for a set of input documents. Our model involves two stages, handling multi-document summarization and generating headline for multi-document. Instead of input user names and destinations in hard-template, our model is able to generate a real customized headline close to the user's preference. Different from TemPEST [3], our Rerank adds user click history to help find the style of template with the user's preference. Since we start directly by selecting the user's favorite template to avoid the problem of sparse input of the encoder, our Rewrite uses a single selective encoder [5].

The proposed model is evaluated on a new dataset from KKday, an e-commerce platform of tourism products. The dataset we used includes product descriptions, product introductions and blog articles. The product introduction data introduces the highlight of products. The product description data detailed introduce the usage and notice for products. Difference between introduction and description is shown in Table 1. Blog articles introduce an attraction which include multiple products related to the attraction. Hence, blog articles and their headlines are the baseline for comparing with the headlines we generated.

**TABLE 1. An example showing difference between introduction and description in our dataset.**

Dataset	Content
Title	American Museum of Natural History Ticket
Introduction	Book with KKday in advance and gain access to the American Museum of Natural History. Admire a world-class collection of around 36 million specimens and cultural artifacts.
Description	<ul style="list-style-type: none"> <li>• <b>Highlights</b> *** Avoid crowds and long lines by booking your tickets to American Museum of Natural History in advance *** Gain entry to both permanent and temporary exhibitions in the American Museum of Natural History *** Explore the American Museum of Natural History, one of the world’s best scientific, educational and cultural institutions.</li> <li>• <b>What You Can Expect</b> ***Beat the lines and crowds by booking your tickets to the American Museum of Natural History with KKday in advance. Step into the Hintze Hall and be greeted by a colossal blue whale skeleton that hangs suspended from the soaring ceilings of the rotunda. From there, explore its exhibition halls and take your time admiring the star specimens and highlights of the museum, including the Alaska Brown Bear, the Great Canoe, Mammoths, and the Willamette Meteorite. Opt for a General Admission + One for a visit to special exhibitions, Space Show, or the IMAX theater. With a Space Show ticket, visit the Hayden Planetarium Space Theater, housed in the top half of the Hayden Sphere. Gaze up at a digital dome projection screen and immerse in a show of hyperrealistic views of planets, star clusters, and galaxies. Skip the lines, save time and energy by booking your American Museum of Natural History tickets with KKday in advance.</li> <li>• <b>Package Info</b> *** General Admission: - Includes 1 American Museum of Natural History Ticket *** General Admission + One: - Includes 1 American Museum of Natural History Ticket + Special Exhibitions / Space Show / IMAX Theater (select 1)</li> <li>• <b>Important Info</b> *** Validity: one day before and after your selected date ***Service Hours: -American Museum of Natural History: 10:00am - 5:45pm -IMAX Theater: 10:30am - 4:30pm (show every hour) -Space Show: 10:30am-4:30pm (show every 1 hour, Wednesday starts at 11:00am) *** All closed on Thanksgiving and Christmas *** Address: Central Park West &amp; 79th St, New York, NY 10024</li> <li>• <b>Additional Info</b> ***Free admission for children under age 2. Please notify the number of children in your party in the 'Notes' section when booking and bring their passport</li> <li>• <b>How to Redeem Your Voucher</b> ***Show your KKday e-voucher at ticketing counter for ticket exchange</li> </ul>

Our contributions are summarized as follows:

- We propose a model MuD2H to generate a headline for a set of documents. This is the first work to use graph neural network to learn representative embeddings which can capture the relationship of sentences across multiple documents. By these embeddings, we extend

the state-of-the-art model for title generation model from single document to generate a headline from multiple documents.

- The proposed model MuD2H utilizes a novel template-based approach, which introduces the soft template as additional input to guide the seq2seq model. The choice of headline template is based on users’ click history data. The headline is then generated based on the sentences embeddings and the selected template by a bi-directional selective encoder.
- To evaluate the proposed model, we create a new dataset for headline generation from multiple documents. This dataset and our implementation details are open for the further research works.<sup>1</sup>
- Experimental result shows our sentence selection method is able to choose key sentences that can keep relevancy and diversity. The proposed model MuD2H can not only outperform other baselines in terms of Rouge scores but also generate a user preferable headline by human evaluation.

The rest of the paper is organized as follows. Section II mentions the past relevant research works, including extractive and abstractive summarization methods, multi-document summarization and two-stage architecture. Section III introduces the proposed model in detail. In section IV, we show the experiment results including baselines, evaluation and case study of our generated headlines. Finally, section V summarizes the conclusion.

## II. RELATED WORK

In this section, we discuss three related topics: extractive summarization, abstractive summarization, and hybrid summarization.

### A. EXTRACTIVE SUMMARIZATION

The Early studies of extract-based summarization were inspired by Pagerank [6]. For example, TextRank [7] and LexRank [8], they computed the salience with a similarity graph of sentences. Li et al. [9] applied the support vector regression model (SVR) for feature selection and weighting to conquer the semantic repeat problem. GreedyKL [10] used Kullback-Lieber (KL) divergence as the criterion for selecting a summary for a given text.

New trends of extractive summarizations have been learning-based. G-Flow [11] works on sentence selection and ordering separately. To evaluate the coherence, G-Flow estimates its quality by using an approximate discourse graph (ADG), based on the hand-crafted feature. R2N2 [12] developed a ranking framework for redundant sentences. It transfers input sentences into a binary tree, then processes the binary tree by RNN recursively at each node. Yasunage et al. [13] applied a graph convolution network [14] onto their proposed personalized discourse graph, and used GRU to calculate

<sup>1</sup>Our dataset and code are available on: <https://github.com/klks0304/mud2h>

sentence embeddings. The model generates cluster embeddings to fully aggregate features between sentences with a document-level GRU. HSG [15] is a model which constructs a heterogeneous graph containing semantic nodes of different granularity levels apart from sentences. Additionally, HSG can flexibly extend from a single document setting to a multi-document setting. SgSum [16] is a graph based method which works on multi-document summarization by extraction based. SgSum consider the sentences as nodes, then generate a sentence relation graph for input documents. The model outputs a subgraph, then considered it as the result of summarization. ThresSum [17] is a recent publish paper applying powerful encoder to emphasize each sentence. The model utilize supervised variables to select sentences as close to original article meaning as possible, without setting limit to select  $k$  sentences.

### B. ABSTRACTIVE SUMMARIZATION

General abstractive summarization approaches have recently shown promising results with sequence-to-sequence neural network architectures [18], which encode documents and then decode the learned representations into an abstractive summary. Rush *et al.* [19] first applied an attention-based sequence-to-sequence model for abstractive summarization. Nallapati *et al.* [20] further changed the sequence-to-sequence model to a fully RNN-based model and achieved outstanding performance. The use of the RNN-based encoder-decoder structure has been used from time to time until now. For example, DRGD [21] uses a recurrent generative decoder to learn latent information of the text.

On the other hand, Cao *et al.* [4] considered that seq2seq models tend to copy source words in order, so they proposed a soft template-based summarization Re<sup>3</sup>Sum. In traditional template-based approaches [1], [22], a template using the manually defined rules is an incomplete sentence which can be filled with the keywords. Because templates are manually defined, it is very time-consuming and also requires a great deal of manual effort. Re<sup>3</sup>Sum [4] proposed a novel soft template-based architecture, which uses existing summaries as templates to guide the seq2seq model. BiSET [23], the state-of-the-art template-based abstractive summarization method, follows the previous architecture. To improve expression for output, BiSET uses a bidirectional selective layer with two gates to select key information. TemPEST [3] proposes a personalized subject generation model, which adds a user-aware sequence encoder to generate user-specific article representation, and assists machine generating user-specific subjects. Most of the abstraction-based summary method are seq2seq model, therefore a toolkit NATS [24] collect these methods and conduct on CNN/Daily Mail dataset. Usually, abstraction-based methods are only suitable for single document summarization. Recent publish method BASS [25] applies semantic graph to connect words in input documents. The method BASS connect inputs can connect words between different documents, therefore it also works on multi-document summarization. BASS [25] successfully

minimize the gap between the multi-document summarization problem and abstract summarization model.

### C. HYBRID SUMMARIZATION

Liu *et al.* [26] proposed a two-stage model T-DMCA for multi-document summarization, concatenating extractive and abstractive summarization methods. This work is not famous by its model, but proposed a well-known dataset, WikiSum, which is applied by following summarization works. T-DMCA has its best result when applying term frequency-inverse document frequency (tf-idf) to rank its sentences in the extractive stage, then applies a transformer decoder with memory-compressed attention in the abstractive stage. HierSumm [27], also a two-stage model, adopts logistic regression to help ranking paragraphs, then applies a global transformer [28] layer to exchange information across paragraphs, and outputs an abstractive summary. ESCA [29] applies a matrix layer after sentence encoder. The matrix layer efficiently controls the outcome of extractor. Since the extracting summarization gives a more human-writing sentence, adjusting the outcome then combines it with the abstractor gives an more high quality summary. TG-MultiSum [30] extract the topic of each document and construct a heterogeneous graph representing each document, then learn for a summary. CABSD [31] works similarly, they extract sentence from the learned subtopic, then generate an abstract summary. The most reason works such as ESCA, TG-MultiSum and CABSD are in two stages. They first extract from input then abstract an output summary, which is the trend of two stages multi-document summarization.

## III. PROPOSED MODEL

Our proposed model is designed in two stages. Before generating a representative headline for input documents, we extract sentences to generate a overall meaning for the documents. In general, our first stage is an extractor and the second stage is an abstractor. Figure 1 shows the structure of the proposed model.

### A. THE EXTRACTOR

In the extractor, given a collection of documents  $D$ , our goal is to extract some salient sentences from these documents. Let  $D$  denote a set of documents as  $D = \{d_i \mid i \in [1, N]\}$ , where  $N$  is the number of documents. Each document  $d_i$  consists of a set of sentences  $S = \{s_{i,j} \mid j \in [1, M_i]\}$ , where  $M_i$  is the number of sentences in  $d_i$ .

Traditional approaches for extraction-based summarization rely on human-crafted features. To adjust this problem, we proposed a data-driven approach, adopting a graph-based learning approach model. We build a sentence relation graph to capture the relation among sentences, each sentence is fed into a recurrent neural network to generate sentence embedding. The next step is to apply the Graph Convolutional Network [14] on the sentence relation graph and sentence embedding as an input node feature. Applying sentence relation graph and sentence embedding on Graph Convolutional

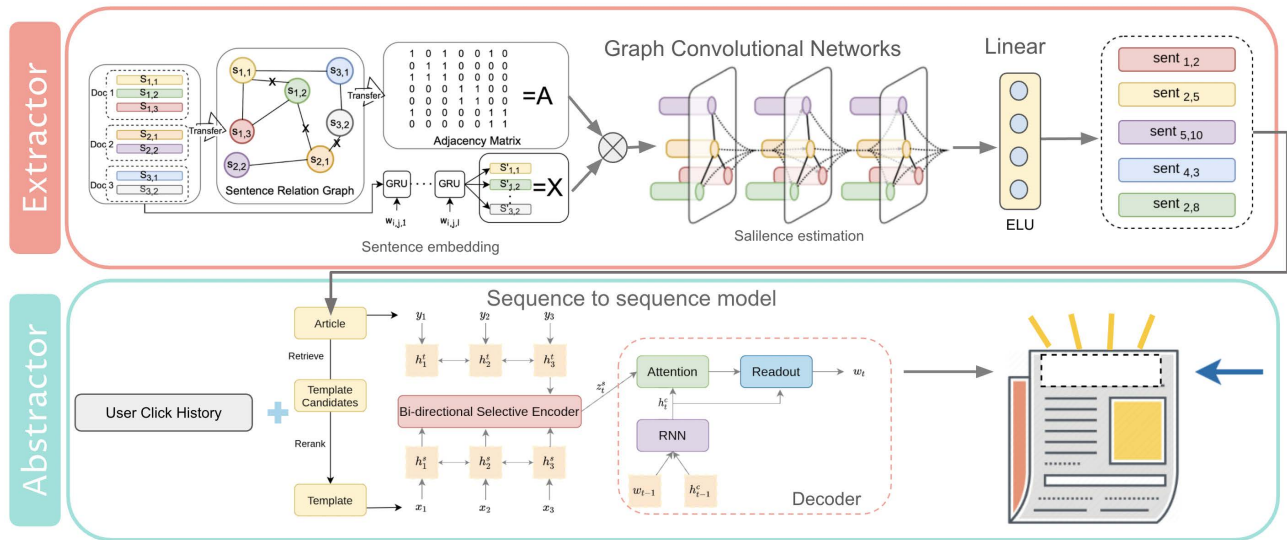


FIGURE 1. The framework of the proposed model MuD2H. There are two stages: extractor and abstractor.

Network can produce a high-level hidden feature for each sentence. After that, we use a linear layer to estimate a saliency score for each sentence. Giving saliency score for each sentence helps model extract suitable sentence from documents. Finally, instead of selecting top saliency score sentences, we use a greedy method to select the saliency sentences to represent input set of documents.

### 1) SENTENCE RELATION GRAPH

In the sentence relation graph, each vertex represents a sentence  $s_{i,j}$ , which means the  $j$ 'th sentence of document  $d_i$ . The weight of the undirected edge between  $s_{i,j}$  and  $s_{i',j'}$  indicates their degree of similarity. We use cosine similarity between each sentence pair  $(s_{i,j}, s_{i',j'})$  and construct a complete graph. However, the model is not able to work significantly if we input this semantic sentence relation complete graph directly, because there is too many redundant information in a complete graph. To emphasize sentences with higher similarity, we set a threshold  $t_g$  and remove the edges that has weight under the threshold. The sentence relation graph is represented as an adjacency matrix  $A$  by graph convolutional network [14] of saliency estimation. The algorithmic form of relation graph generating process is given in Algorithm 1.

### 2) SENTENCE EMBEDDING

Given a collection of documents  $D$ , we encode all sentences which have appear in each document. For all words in sentence  $s_{i,j}$ , we convert each word into a word embedding, then feed word embeddings in a sentence  $s_{i,j}$  into the sentence encoder to generate  $s_{i,j}$ 's sentence embedding  $s'_{i,j}$ . The dimension of sentence embedding  $s'_{i,j}$  is  $d_s$ . We use a recurrent neural network (RNN) with Gate Recurrent Unit (GRU) as the sentence encoder, where the last hidden state is sentence embedding. All sentence embedding from the given

### Algorithm 1: Sentence Relation Graph

**Input:** A set of documents  $D = \{Doc_1, Doc_2, \dots, Doc_N\}$ .  
**Output:** Sentence relation Graph  $G$ .  
 Let the set of sentences be  $S = \{s_{1,1}, \dots, s_{N,M_N}\}$ ;  
 Construct a complete graph  $G$  with  $M$  nodes;  
 Each nodes represent a sentence in  $S$ ;  
**for**  $s_{i,j}, s_{i',j'} \in S$  **do**  
     **if**  $similarity(s_{i,j}, s_{i',j'}) < t_g$  **then**  
         Delete edge  $(s_{i,j}, s_{i',j'})$   
     **end**  
     edge weight =  $similarity(s_{i,j}, s_{i',j'})$   
**end**

collection of documents are concatenated as the following:

$$X = [s'_{1,1} \dots s'_{i,j}]^T \in \mathbb{R}^{M \times d_s} \quad (1)$$

Note that  $M = \sum_{i=1}^N M_i$  represents the number of all sentences have appears in the document set  $D$ . The matrix  $X$  will be considered as the feature matrix to apply the graph convolutional network [14] using saliency estimation.

### 3) SALIENCE ESTIMATION

A Graph Convolutional Network is a multi-layer neural network which operates directly on a graph and induces embedding vectors of nodes based on properties of their neighborhoods. Layer-wise linear formulation allows the model to capture higher level hidden feature in sentences. We use adjacency matrix  $A$  to formulate sentence graph, and use  $X$  as its feature matrix representing in this step.

- $A \in \mathbb{R}^{M \times M}$ , the adjacency matrix of the sentence relation graph, where  $M$  is the number of vertices. In particular, if the  $i$ 'th node is adjacency to the  $j$ 'th node, then  $a_{ij} = 1$ . Otherwise,  $a_{ij} = 0$ .

- $X \in \mathbb{R}^{M \times d_s}$ , the input node feature matrix, where  $d_s$  is the dimension of feature vectors.

The output of this stage is a high-level hidden feature for each node,  $S'' \in \mathbb{R}^{M \times F}$ , where  $F$  is the dimension of output vector embedding. In order to include the nodes' own features in the aggregate, we add self-loops to the adjacency matrix  $A$  such that  $\hat{A} = A + I_M$ , where  $I_M$  is the identity matrix. Our propagation rule follows:

$$S'' = \text{ELU}(\hat{A} \cdot \text{ELU}(\hat{A}XW_0 + b_0)W_1 + b_1) \quad (2)$$

$\hat{A} = D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}$  is the normalized symmetric adjacency matrix.  $D$  is the degree matrix where its  $i$ th diagonal elements is sum of elements in  $i$ th row of  $\tilde{A}$ .  $W_l$  is an input-to-hidden weight matrix to learn in the  $l$ th layer, and  $b_l$  is the bias vector. We use Exponential Linear Unit (ELU) [32] instead of Reflect Linear Unit (ReLU) [33] as the activation function, because Exponential Linear Unit tends to converge cost to zero faster and deal with the vanishing gradient problem better. Subsequently, we use a linear layer to project the high-level hidden feature of each sentence to the salience score. Additionally, we normalize the salience score via softmax:

$$\text{sal}(s_{i,j}) = \text{softmax}(s''_{i,j}W_2 + b_2) \quad (3)$$

Note that  $s_{i,j}$  is the  $j$ th sentence in  $i$ th document, and  $s''_{i,j}$  is the  $(\sum_{i'=1}^j M_{i'} + j)$ th row of  $S$  with  $M_0 = 0$ .

#### 4) TRAINING

Previous works [13], [34] use cross-entropy loss for training. When we trained our model with cross-entropy, the loss tends to output scores close to 0 or 1 which may cause an obstacle for ranking. To overcome this problem, we trained the model with contrastive loss. Since we select sentences by salience score, sentence selection problem can be considered as a ranking problem. The relative ranking between sentences is considered more important than absolute scores, and contrastive loss gives the relative ranking score. Thus, referring to contrastive loss [35], we define the ranking loss as:

$$\mathcal{L} = \frac{1}{2} \sum_{s_i, s_j \in S} \left( (1 - y) \cdot D(\text{sal}(s_i), \text{sal}(s_j))^2 + y \cdot \max(\mu - D(\text{sal}(s_i), \text{sal}(s_j)), 0)^2 \right) \quad (4)$$

where  $y$  is a label, which represents whether the rankings of the two sampled sentences are similar ( $y = 0$ ) or far ( $y = 1$ ) comparing with  $\sigma = \sqrt{\text{Var}(R(S))}$ .  $\mu$  is a setting margin.  $D$  represents the distance between two given sentence, we use Euclidean distance here. More precisely,

$$D(x_i, x_j) = \|x_i - x_j\|. \quad (5)$$

and

$$y = \begin{cases} 0, & \text{if } D(R(s_i), R(s_j)) \leq \sigma \\ 1, & \text{if } D(R(s_i), R(s_j)) > \sigma \end{cases} \quad (6)$$

$R(s) = \text{softmax}(r(s))$ , where  $r(s)$  is the ROUGE-1 recall score of sentence  $s$  by measuring with the ground-truth. The

objective function represents that if two data points are considered similar ( $y = 0$ ), we minimize the distance between them. Far pairs contribute to the loss function only if their distance is within a specified margin. When the distance between two data points is considered far ( $y = 1$ ) and their distance is less than the margin, we replace their distance as the margin, to let the loss function give a penalty.

#### 5) SENTENCE SELECTION

After sorting the sentences in descending order according to the predicted scores of our model, we start to choose sentences. Rather than intuitively selecting the Top-k sentences, we apply a greedy strategy to select sentences. Greedy strategy is able to select diversity sentences instead of repeated meaning sentences [12], [36]. Every time we select one sentence from the top of the list, we check whether it is non-redundant with the existing sentences. To determine whether the sentence is redundant, we use tf-idf cosine similarity. For an input sentence  $s$  and selected sentence set  $C$ , if cosine similarity between  $s$  and all sentences in  $C$  is small, and the sentences already selected is above a threshold  $t_s$ , the sentence is considered redundant. If not, we select the sentence. We repeat this step until the expected number of sentences  $n$  is reached. **red**The algorithmic form us shown in Algorithm 2.

---

#### Algorithm 2: Sentence Selection Algorithm

---

**Input:** Embedded sentences  $S' = \{s'_{1,1}, \dots, s'_{N,M_N}\}$

**Output:** Set of selected sentence  $C$ .

Sort  $S'$  by  $\text{sal}(s_{i,j})$  descending.;

Let sorted  $S' = [s'_1, s'_2, \dots, s'_M]$ ;

$C = \{s'_1\}$ ;

**for**  $s \in S'$  **do**

**for**  $t \in C$  **do**

**if**  $\text{similarity}(s, t) > t_s$  **then**

            Drop  $s$ ;

**end**

**end**

**if**  $s \neq \text{none}$  **then**

        Add  $s$  to  $C$ ;

**end**

**if**  $|C| > n$  **then**

**return**  $C$

**end**

**end**

**return**  $C$

---

#### B. THE ABSTRACTOR

In the abstractor, our goal is to generate a headline, which needs to be personalized, attractive, faithful and within the length constraint. Therefore, we referred to previous template-based summarization [4], [23] frameworks in the abstractor. The input of the abstractor is a collection of sentences produced by the extractor. These sentences were

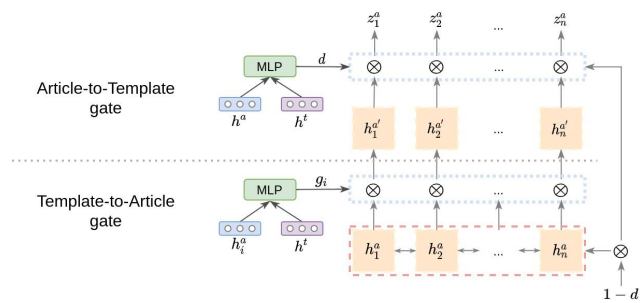


FIGURE 2. The selective encoder.

concatenated, hence it can be considered as an article. Each article  $A_r$  consists of  $n$  words  $\{x_i^a \mid i \in [1, n]\}$ . Let  $T$  denote a set of templates in the training corpus as  $T = \{t_i \mid i \in [1, p]\}$ , where  $p$  is the number of all template candidates in our dataset.

For the given article, we use the Information Retrieval (IR) platform to find out some soft template candidates from  $T$ , and then further choose the best template  $T' = (x_1^t, x_2^t, \dots, x_n^t)$  by Rerank or user click history. Subsequently, we extend a seq2seq model to generate a headline by learning important information from  $A_r$  and  $T'$ .

### 1) RETRIEVE AND RERANK

The goal of Retrieve and Rerank is choosing the best template  $t$  for  $A_r$ . Retrieve aims to return some template candidates from the training corpus. We assume that similar sentences hold similar summary patterns. Therefore, given an article, we find its analogy in the training corpus and pick their headlines as the template candidates. Given  $A_r$ , we use the widely-used IR system PyLucene<sup>2</sup> to retrieve a set of similar articles, and their headline will be treated as the template candidates. For each  $A_r$ , we choose the top 30 searching results as template candidates.

The Retrieve process is only based on word matching or text similarity, but does not measure their deep semantic relationship. Therefore, we use Doc2Vec [37] embedding to compute cosine similarity to identify the best template in the template candidates. Additionally, in our work, we expect our generating headline to be personalized, so we add the user click history to help us choose a template. We record the title of the product that the user has clicked as user click history. As a result, in the Rerank process, we join the user click history to compute cosine similarity with template candidates to select our desired template  $t$  for  $A_r$ .

### 2) REWRITE

Our implementation model in the Rewrite step is inspired by BiSET [23] and selective mechanism [5]. Before the Rewrite step, remind that we have a source article  $A_r$  and its suitable template  $T'$  learned from Retrieve and Rerank. We use a two-layer Bidirectional Long Short-Term

Memory (BiLSTM) as the encoder layer to encode the article and the template into hidden states  $h_i^a$  and  $h_i^t$  respectively. The role of Rewrite is to select important information. As shown in Figure 2, there are two selective gates: the Template-to-Article (T2A) gate and the Article-to-Template (A2T) gate. The T2A gate can apply the template to filter the article representation. We concatenate the last forward hidden state  $\overrightarrow{h}_n^t$  and backward hidden state  $\overleftarrow{h}_1^t$  as the template representation  $h^t$ . For each time step  $i$ , it takes  $h^t$  and  $h_i^a$  as inputs to output a template gate vector  $g_i$  to select from  $h_i^a$ :

$$g_i = \sigma(W_a h_i^a + W_t h^t + b_a) \quad (7)$$

$$h_i^{a'} = h_i^a \odot g_i \quad (8)$$

where  $\sigma$  denotes the sigmoid activation function, and  $\odot$  is element-wise multiplication. After the T2A gate, we obtain a sequence of vectors  $(h_1^{a'}, h_2^{a'}, \dots, h_n^{a'})$ .

The goal of the A2T gate is to control the proportion of  $h^{a'}$  in final article representation. We assume that the source documents are credible, therefore implies current stage article  $A_r$  is credible, and learn a confidence degree  $d$  to decide the proportion of  $h_i^{a'}$ :

$$d = \sigma((h^a)^T W_d h^t + b_d) \quad (9)$$

$h^a$  is generated in the same way as  $h^t$ : concatenating the forward hidden state  $\overrightarrow{h}_n^a$  and backward hidden state  $\overleftarrow{h}_1^a$ .

The final article representation is computed by the weighted sum of  $h_i^{a'}$  and  $h_i^a$ :

$$z_i^a = d \cdot (h_i^{a'}) + (1 - d) \cdot (h_i^a) \quad (10)$$

The above finishes the encoding part of the input article, it selects important information then gives a vector representation. In the decoder part, we stacked two layers of an Recurrent Neural Network with a Long Short-Term Memory unit, and use an attention mechanism [38] to generate the headline. At each time step  $t$ , LSTM reads the previous word embedding  $w_{t-1}$  and hidden state  $h_{t-1}^c$  generated in the previous step, and then outputs a new hidden state for the current step:

$$h_t^c = LSTM(w_{t-1}, h_{t-1}^c) \quad (11)$$

where the initial hidden state of the LSTM is the original article representation  $h^a$ .

The context vector  $c_t$  for current time step  $t$  is computed through the concatenate attention mechanism [38], which uses  $h_t^c$  and  $z^a$  to get importance scores. The importance scores are then normalized to get the current context vector by weighted sum:

$$c_t = \sum_{i=1}^L a_{t,i} z_i^a \quad (12)$$

$$a_{t,i} = \frac{\exp(e_{t,i})}{\sum_{i=1}^L \exp(e_{t,i})} \quad (13)$$

$$e_{t,i} = (z_i^a)^T W_c h_t^c \quad (14)$$

<sup>2</sup><https://lucene.apache.org/pylucene/>

Subsequently, we use a concatenation layer to combine the hidden state  $h_t^c$  and context vector  $c_t$  into a new readout hidden state  $h_t^o$ :

$$h_t^o = \tanh(W_o[c_t; h_t^c]) \quad (15)$$

In the final stage,  $h_t^o$  is fed into a softmax layer to output the target word distribution for predicting the next word  $w_t$  over existing words  $w_1, w_2, \dots, w_{t-1}$ :

$$p(w_t | w_1, \dots, w_{t-1}) = \text{softmax}(W_p h_t^o) \quad (16)$$

### 3) TRAINING

The objective function includes two parts. To learn the generation of headlines, we minimize the negative log-likelihood between the generated headline  $w$  and the human-written headline  $w^*$ :

$$\mathcal{L}_h = -\frac{1}{M} \sum_{i=1}^M \sum_{j=1}^L \log p(w_j^{*(i)} | w_{j-1}^{(i)}, x^{a(i)}, x^{t(i)}) \quad (17)$$

To learn the style of the template, we minimize the negative log-likelihood between generated headline  $w$  and the template  $w^f$ :

$$\mathcal{L}_t = -\frac{1}{M} \sum_{i=1}^M \sum_{j=1}^L \log p(w_j^{(i)} | w_{j-1}^{(i)}, x^{a(i)}, x^{t(i)}) \quad (18)$$

In other words, adjusting  $\mathcal{L}_h$  optimize the capture information from input documents. If  $\mathcal{L}_h$  is small, then it is close to the original meaning of the input sets. On the other hand, adjusting  $\mathcal{L}_t$  optimize the personalized style of the headline. When  $\mathcal{L}_t$  is small, it is closer to user's favor template, therefore outputs a personalized headline. The final objective function combines the above two:

$$\mathcal{L} = \mathcal{L}_h + \alpha \mathcal{L}_t \quad (19)$$

## IV. EXPERIMENT

The goal of this work is to generate a suitable headline for input set of documents. More specifically, we convert our problem into the following questions:

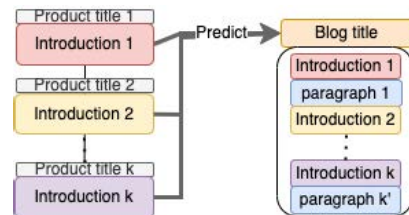
- How can the sentence relation graph be constructed in order to achieve the model's best performance?
- Can our extractor outperform other extraction-based summarization models?
- Is using the complete two-stage architecture model better than just using the abstractor?

### A. DATASETS

We used a real-world dataset provided by the traveling e-commerce platform, KKday.<sup>3</sup> KKday provides over 30,000 products from over 90 countries, including local tours, activities, and tickets. We trained the extractor using product introductions, descriptions and titles. The KKday blog dataset that mentions that different products provide materials for

<sup>3</sup><https://www.kkday.com/zh-tw>

research on MuD2H applications in multiple documents. On average, each blog mentioned eight products. The headlines generated by MuD2H were compared with the original headline of the blog article. In conclusion, 80% of the dataset was assigned for training, and 20% for validation and testing. Figure 3 describes the relationship between product introductions and blog information. Dataset and implement detail are provide in the supplementary material.<sup>4</sup> Overview of our dataset is describe in Table 2.



**FIGURE 3.** The relation between product introduction and blog data. A blog's title is consider as a title for these k product's multi-document title.

**TABLE 2.** An overview of our dataset.

	Document with headline	Sentences
<b>Blog</b>	1813	87377
Introduction	31451	62004
Description	66215	321088

### B. IMPLEMENTATION DETAILS

To set the edge weight of our sentence relation graph, we set  $t_g = 0.1$  according to the following experiment. Each document is tokenized by Chinese Knowledge and Information Processing (CKIP). Word2Vec [39] and Doc2Vec [37] embedding is implemented with gensim and pretrained on the latest Chinese Wikipedia dataset. The output dimension of sentence embeddings is the same as word embedding, i.e. 250. For the graph convolutional network, we set the embedding size of the first convolution layer as 400 and the embedding size of the second convolution layer as 128. The batch size we use is 16. The objective function is optimized using Adam [40] stochastic gradient descent with a learning rate of 0.0075 and early stopping with a window size of 10. We apply dropout with probability 0.2 before the linear layer. The threshold  $t_s$  in sentence selection is 0.8 (tuned on validation set). For the abstractor, we construct our architecture referring to BiSET [23], which is extended from the popular seq2seq framework OpenNMT [41]. The size of word embeddings and LSTM hidden state are set to 500. Additionally, the objective function is optimized using Adam optimizer with a learning rate of 0.001. For all baseline models, we use default parameter settings in their original paper or implementation.

### C. EVALUATION METRICS

To analyze the influence of the different methods in the sentence relation graph, we use Normalized Discounted Cumulative Gain (NDCG) [42] for evaluation. NDCG is a

<sup>4</sup><https://github.com/kllks0304/mud2h>

**TABLE 3.** Analysis of sentence relation graph construction with NDCG@6.

Method	LSTM			GRU		
	0.0	0.1	0.2	0.0	0.1	0.2
Cosine	0.52	0.75	0.63	0.58	<b>0.77</b>	0.72
TextRank	0.36	0.41	0.43	0.44	0.53	0.41
LexRank	0.51	0.40	0.50	0.45	0.45	0.59
Tf-idf	0.40	0.49	0.44	0.47	0.43	0.52

ranking evaluation metric. We view our problem as ranking problem in training the extractor, so we use NDCG for performance comparison.

For the summarization task, we adopt Rouge [43] score for automatic evaluation. Rouge-1 and Rouge-2 are the rate of the length of the largest common sub-sequence, and Rouge-L can find out the longest common sub-sequences of words between the original summary and the predicted summary. Additionally, we use Word2Vec [39] cosine similarity to measure the average similarity between the output and each document because we expect that our output can express the meaning of each document.

#### D. SENTENCE RELATION GRAPH COMPARISON

Different methods converting relations between sentences into numeric result will influence our sentence relation graph. We try different ways including two embedding methods (LexRank and TextRank). Convert the value of  $t_g$  from 0 to 0.2 to observe the impact. The considered methods include:

- 1) Cosine: Calculate Word2Vec cosine similarity between each sentence pair.
- 2) TextRank [7]: A weighted graph is created where nodes are sentences and edges defined by similarity measures based on word overlap. Then we use an algorithm similar to PageRank [6] to calculate the importance of the sentence and the precise edge weight. The transition matrix that describes the Markov chain used in PageRank is extracted.
- 3) LexRank [8]: A widely used multi-document extractive summarizer based on the concept of eigenvector centrality in a graph of sentences is used to set up the edge weights. We build a graph with sentences as nodes and edges weighted by tf-idf cosine similarity, then run a PageRank-like algorithm.
- 4) tf-idf: Consider a sentence as query and all the sentences in multi-document as the document. The weight corresponds to the cosine similarity between each query pair.

Table 3 is the experiment result. We choose the best method and parameters of the experimental results for the rest of MuD2H model (our model). The result shows that using cosine similarity to build the sentence relation graph is significantly better than other methods on NDCG evaluation. The possible reason is that cosine similarity relies on the semantics of the sentences rather than its words matching.

**TABLE 4.** Rouge recall scores for various extraction-based models on the test set.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Random	31.26	11.70	21.99
<b>Top-k</b>	42.11	15.51	26.01
TextRank [7]	38.69	14.61	25.90
Cont. LexRank [8]	37.28	14.45	23.93
MMR [46]	39.96	15.22	26.15
SemSentSum [34]	40.62	15.64	<u>27.22</u>
<b>Ours(extractor)</b>	<b>42.62</b>	<b>16.50</b>	<b>27.89</b>

**TABLE 5.** Average Word2Vec cosine similarity and standard deviation between the output of the extractor and each document.

Model	Similarity
TextRank [7]	0.7498±0.0577
Cont. LexRank [8]	0.7322±0.0586
MMR [46]	0.7517±0.0568
SemSentSum [34]	0.7502±0.0565
<b>Ours(extractor)</b>	<b>0.7533±0.0571</b>

#### E. QUALITATIVE RESULTS

First, we compared our extractor model with some extraction-based summarization. Table 4 presents the results of the ROUGE recall scores. Random represents randomly choosing k sentences in our sentence selection set, and Top-k takes the top similar sentence by cosine similarity. Compared with traditional methods, for example TextRank [7] and Continuous LexRank (Cont. LexRank) [8], our model performed better in the Rouge score. The state-of-the-art graph-based approach SemSentSum [34] is a fully data-driven model that uses cross-entropy as the objective function. As expected, it outperformed other traditional baselines in Rouge-2 and Rouge-L, but our model still performed better. This is because sentence ranking starts to become unstable in the deeper layer because SemSentSum applies cross-entropy as their objective function, loss tends to fade and our contrastive loss function plays a role. Maximal Margin Relevance (MMR) [44] is a well-known greedy algorithm for multi-document [45], and improvements of MMR have been proposed. For comparison, we use state-of-the-art phrase embedding-based MMR [46] as a baseline. It focuses on producing a non-redundant summary, so its output has relatively high word diversity. The Rouge-1 score of Top-k is higher than Rouge-2 and Rouge-L scores. It can be seen that these scores of Top-k are close to those of the proposed method, which means Top-k can also include sentences with close meaning. However, Table 6 presents a case study to demonstrate the limitation of Top-k. In brief, Top-k selects sentences with repeated meaning. As shown in Table 6, the first and second sentence selected by Top-k are about the Universal Express Pass. On the contrary, the proposed method could select sentences by taking diversity and relevancy into account. This result shows the advantage of the proposed method. However, it is challenging to determine whether finding diverse sentences is the key because more off-topic sentences could be found. In terms



**TABLE 6.** Case study for showing advantages of our sentence select method. Each of the case contains seven sentences. In top-k, the second selected sentence is repeated, because it has similar meaning with the first one. Avoiding repeated sentences can make the model capture more meaning. (English version are translated from Chinese).

method	example
Top-k	<ul style="list-style-type: none"> <li>日本大阪環球影城™標準4飛天翼龍4快速@通關券KKday官網同步開賣(Universal Express Pass 4: Premium or The Flying Dinosaur)</li> <li>日本大阪環球影城™標準7飛天翼龍7逆轉世界7快速@通關券KKday官網同步開賣(Universal Express Pass 7 - Standard, The Flying Dinosaur, or Backdrop)</li> <li>VIP手環提早入園享受專屬通關 (Early park enter,enjoying exclusive pass with VIP wristband)</li> <li>省下多次轉車時間搭乘直Q巴士從京都直達環球影城用輕鬆方式往返樂園(Save transferring time by taking Q bus from Kyoto to Universal Studio Japan, an easy round trip to the park)</li> <li>立即KKday預訂大阪環球影城VIP手環門票在阿倍野16樓售票口領取直接入場(Immediately use KKday to book the Universal Studio Japan VIP wristband ticket, take your ticket and enter at Abeno 16F ticket counter)</li> <li>住在京都旅客不用煩惱抵達日本環球影城™交通(Don't need to worry about the transportation to Universal Studio Japan if you stay in Kyoto)</li> <li>快速@通關券可以保證入園哈利波特魔法世界™及七種遊樂設施快速通關省去抽整理券時間讓輕鬆暢遊日本環球影城 (Express pass can ensure you enter the Harry Potter's Magical world and 7 rides fast passing saving your time in collecting number plate, easy enjoying in Universal Studion Japan.)</li> </ul>
ours	<ul style="list-style-type: none"> <li>日本大阪環球影城™標準4飛天翼龍4快速@通關券KKday官網同步開賣 (Universal Express Pass 4: Premium or The Flying Dinosaur)</li> <li>VIP手環提早入園享受專屬通關(Early park enter,enjoying exclusive pass with VIP wristband)</li> <li>省下多次轉車時間搭乘直Q巴士從京都直達環球影城用輕鬆方式往返樂園(Save transferring time by taking Q bus from Kyoto to Universal Studio Japan, an easy round trip to the park)</li> <li>立即KKday預訂大阪環球影城VIP手環門票在阿倍野16樓售票口領取直接入場Immediately use KKday to book the Universal Studio Japan VIP wristband ticket, take your ticket and enter at Abeno 16F ticket counter)</li> <li>住在京都旅客不用煩惱抵達日本環球影城™交通(Don't need to worry about the transportation to Universal Studio Japan if you stay in Kyoto)</li> <li>快速@通關券可以保證入園哈利波特魔法世界™及七種遊樂設施快速通關省去抽整理券時間讓輕鬆暢遊日本環球影城(Express pass can ensure you enter the Harry Potter's Magical world and 7 rides fast passing saving your time in collecting number plate, easy enjoying in Universal Studion Japan.)</li> </ul> <p>UniversalCoolJapan日本環球影城™年限定動漫主題新穎有趣活動受到國內外遊客廣大迴響今年1月18日開始6月23日限定推出不僅最熱門《名偵探柯南》更多如《魯邦三世》這部經典動漫內容結合解謎餐廳賞景遊樂設施真實逃脫遊戲等讓現場感受極致興奮遊樂設施身為動漫迷的你千萬不能錯過</p>

of results, graph-based methods including our model and SemSentSum are better than MMR in ROUGE recall score.

In the multi-document summarization task, an important goal is that the generated results need to express the focus

**TABLE 7.** Compare the results of different extractors adding the abstractor with Rouge F1 scores and average cosine similarity.

Model	Rouge-1	Rouge-L	Similarity
None	17.87	15.83	0.5010
TextRank [7]	18.14	16.52	0.5160
Cont. Lexrank [8]	18.08	16.53	0.5201
MMR [46]	18.10	16.90	0.5266
SemSentSum [34]	18.41	16.90	0.5241
<b>MuD2H</b>	<b>19.05</b>	<b>16.96</b>	<b>0.53</b>

**TABLE 8.** Ranking result by human evaluation. Average represent the average ranking.

Model	Sample 1	Sample 2	Sample 3	Average
MMR	2.839	3.548	2.839	3.075
SemSentSum	2.903	2.742	2.935	2.860
MuD2H	2.419	2.096	2.483	2.333
Human-written	<b>1.839</b>	<b>1.548</b>	<b>1.742</b>	<b>1.710</b>

of each document. This problem is at the semantic level, so we adopt Word2Vec similarity. We measure the average similarity and standard deviation between our outputs and each input document. The average similarity should be as large as possible, but the standard deviation should be as small as possible. Table 5 shows the cosine similarity for different models. Our model has the highest average cosine similarity. Since our input set of documents have clear relation, for example, products mentioned in the same blog must from same city, therefore a success multi-document model should at least catch the city characteristic. If the model catches the common characteristic, it is easy to get high score in our experiment results. In other words, it is difficult to get low scores for the models we select.

In order to prove that our two-stage model is useful, we separately use the output of the different extractors and the result of directly concatenating the documents as the input of the model. Table 7 shows the result of the experiment. We use Rouge F1 scores between our generated headline and human written headline, and average Word2Vec cosine similarity between our generated headline and every document. The performance of our model is better when we use the extractor in the first stage. We consider that this is due to the fact that the extractor has the focus of capturing cross-documents. Furthermore, our complete model beats all the baseline models. It shows the best result on real dataset application.

## F. QUANTITATIVE RESULTS

### 1) HUMAN EVALUATION

In addition to the automatic evaluation, we also access model performance by human evaluation in a real case. We conducted a user survey with 31 users, including computer science graduate students and web users. Each sample includes a set of product introductions and headlines generated by different methods. We ask the users to rank each headline on a scale of 1 to 4. The result in Table 8 shows that the most attractive headline is human-written, and the second place is generated by our model. In our statistics, 65% of people consider that human-written headlines are the best, and 50%

**TABLE 9.** Input example of the personalized headline generated by our full model. (English version are translated from Chinese).

Document 1	宜蘭綠舞座落於宜蘭五結，交界蘭陽平原，鄰近蘭陽溪、宜蘭河及冬山河，更可觀望龜山島海景，為全台唯一兼具景觀、人文藝術、生態與休閒遊憩的日式庭園渡假園區。透過KKday預訂浴衣與抹茶體驗，無論大人小孩不用到日本也能在這感受濃濃的日式風情。東部網美秘境 ("DANCEWOODS-Yilan" is located in Wujie Township, Yilan Taiwan which has the gorgeous landscape of Lanyang Plain. DANCEWOODS-Yilan is near the intersection of three rivers, Lanyang, Yilan and Dongshan, with splendid view of Guishan Island. Dancewoods-Yilan is the first and the only Japanese garden-themed resort hotel in Taiwan. Book yukata and matcha experience with KKday, or Ninja no Mori experience camp, so adults and children can enjoy a fantastic Japanese style vacation here without having to go to Japan.)
Document 2	現在預訂KKday宜蘭賞鯨龜山島半日遊，4小時內搭船賞鯨或環島或登島。龜山島登島由專業嚮導帶領，帶你從烏石港搭船輕鬆出發，登上太平洋上的神秘仙島—龜山島，一覽龜山島八景！另可選擇每日限量名額401高地攻頂挑戰，帶你登上台灣離島第二高山。(Enjoy a scenic boat ride along Yilan's coast, while watching dozens of racing dolphins and whales. Explore Guishan Island, where Taiwan's only active volcano is located.)
Document 3	立即預訂KKday龜山島賞鯨套票，一人成行無需煩惱任何交通問題，乘船前往宜蘭賞鯨，近距離感受海豚的熱情，也可選擇登上龜山島一覽其美麗景色，結束後可自行安排至頭城老街、幾米公園等景點行程，感受不一樣的宜蘭之旅。(Board a boat sailing towards Guishan Island. Seize the chance to get close to dolphins and whales. Opt for a short tour around Gueishan Island and appreciate its stunning oceanic views.)
Document 4	立即訂購KKday宜蘭帆船輕旅行，搭乘重型帆船看見不一樣的龜山島，喚醒您的海洋魂！在船上由專業船員指導，體驗當水手及船長的滋味，挑戰操帆、結繩、跳水等水上活動，一邊欣賞一望無際的蔚藍太平洋，一邊吹拂涼爽海風，享受一趟海派半日遊！(Yilan sailing boat jaunt, riding a sailing boat to see a different Guishan Island, which to wake your ocean spirit! A professional guide will lead you on the sailing boat. Experience being a sailor and a captain, controlling the sail, tying knots, diving and other water activities. Enjoy the endless blue Pacific Ocean, face the warm sea breeze, immerse yourself in an "oceanful" half day tour!)
Document 5	上帝在龜山島打翻了一杯牛奶，夢幻仙境牛奶海就此誕生！即刻預訂KKday龜山島牛奶海歡樂巡航玩水之旅，近距離欣賞美麗的牛奶海，在安全海域體驗水上活動讓你玩的最盡興。夏天必朝聖的網美行程，去年錯過的今年一定要跟上！(God overturned a glass of milk on Guishan Island, and the dreamy wonderland milk sea was born! Book now for a KKday Guishan Island Milk Sea Happy Cruise Water Tour, enjoy the beautiful Milk Sea up close, and experience water activities in the safe sea to give you the greatest enjoyment. A pilgrimage to Net Beauty in the summer is an experience not to be missed.)
Real Subject	【宜蘭龜山島旅遊】2020 龜山島景點、登島/繞島/環島行程、龜山島牛奶海/賞鯨豚，龜山島一日遊這樣玩就對！(Yilan Guishan Island Tourism: Guishan Island Attractions & Landing / Around the island / Itinerary around the island & Guishan Island Milk Sea / Whale watching, This is the right way to go for a day trip to Guishan Island !)

**TABLE 10.** Result of the personalized headline generated by our full model. (English version is translated from Chinese).

User1	【宜蘭一日遊】福隆海濱公園、獨木舟體驗 (Yilan One Day Tour: Fulong Seaside Park & Canoe experience)
User2	【宜蘭樂園】朝灣獨木舟體驗(Yilan Paradise: Chaowan Kayak Experience)
User3	【宜蘭必去景點】福隆門票+獨木舟體驗 (Must-visit attractions in Yilan: Fulong Ticket + Canoe experience)

of people consider that the headlines generated by our model are second only to the human-written. However, our model is the best over these auto-generated headlines.

2) CASE STUDY

Table 9 and Table 10 shows a case study of the customized headline generation task. Given multi-product introductions as Table 9, our proposed model can generate different style headline according to different template as Table 10. Users may favor in different template, therefore attract by different headline. We design the user-specific headlines according to the click history of other products.

V. CONCLUSION

In this study, we propose a two-stage model MuD2H that generates a summary and headline for multiple documents. To the best of our knowledge, this is the first model to generate headlines for multiple documents. To evaluate the proposed model MuD2H, we collect a new dataset from an e-commerce site of tourism products, which contained product descriptions, product introductions, blog articles, and user browsing records. The first stage of our research involved graph-based extractive summarization. We applied a graph convolutional network to learn the sentence features for salience estimation. Our cross-calculation ensures that the output summary covers the meanings of the input document set, rather than repeating words or sentences. The second stage is template-based abstractive summarization. We learn users' text preferences from their browsing history and then apply their favorite headline type as a soft template to guide the seq2seq model. MuD2H outperforms the existing summarization models and meets the company's requirement of generating personalized headlines for different users. In addition, we present human evaluations and case studies to illustrate our results.

ACKNOWLEDGMENT

The authors appreciate KKday's data group for providing dataset. They would like to thank Yu-Chien Tang's help in experiments.

REFERENCES

[1] L. Zhou and E. Hovy, "Template-filtered headline summarization," in *Proc. Int. Work. Text Summarization Branches Out (ACL)*, 2004.  
 [2] T. Liu, H. Li, J. Zhu, J. Zhang, and C. Zong, "Review headline generation with user embedding," *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. New York, NY, USA: Springer, 2018, pp. 324–334.

- [3] Y.-H. Chen, P.-Y. Chen, H.-H. Shuai, and W.-C. Peng, "TemPEST: Soft template-based personalized EDM subject generation through collaborative summarization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 7538–7545.
- [4] Z. Cao, W. Li, S. Li, and F. Wei, "Retrieve, rerank and rewrite: Soft template based neural summarization," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 152–161.
- [5] Q. Zhou, N. Yang, F. Wei, and M. Zhou, "Selective encoding for abstractive sentence summarization," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1095–1104.
- [6] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. SIDL-WP-1999-0120, 1999.
- [7] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 404–411.
- [8] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, Dec. 2004.
- [9] S. Li, Y. Ouyang, W. Wang, and B. Sun, "Multi-document summarization using support vector regression," in *Proc. DUC*, 2007, p. 42.
- [10] A. Haghighi and L. Vanderwende, "Exploring content models for multi-document summarization," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2009, pp. 362–370.
- [11] J. Christensen, S. Soderland, and O. Etzioni, "Towards coherent multi-document summarization," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2013, pp. 1163–1173.
- [12] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, "Ranking with recursive neural networks and its application to multi-document summarization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 29, 2015, pp. 1–7.
- [13] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev, "Graph-based neural multi-document summarization," in *Proc. CoNLL*, 2017, p. 452, 2017.
- [14] N. Thomas Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.
- [15] D. Wang, P. Liu, Y. Zheng, X. Qiu, and X. Huang, "Heterogeneous graph neural networks for extractive document summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6209–6219.
- [16] M. Chen, W. Li, J. Liu, X. Xiao, H. Wu, and H. Wang, "SgSum: Transforming multi-document summarization into sub-graph selection," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 4063–4074.
- [17] R. Jia, Y. Cao, H. Shi, F. Fang, P. Yin, and S. Wang, "Flexible non-autoregressive extractive summarization with threshold: How to extract a non-fixed number of summary sentences," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 13134–13142.
- [18] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2, 2014, pp. 3104–3112.
- [19] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 379–389.
- [20] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using Sequence-to-sequence RNNs and beyond," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 280–290.
- [21] P. Li, W. Lam, L. Bing, and Z. Wang, "Deep recurrent generative decoder for abstractive text summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2091–2100.
- [22] L. Wang and C. Cardie, "Domain-independent abstract generation for focused meeting summarization," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2013, pp. 1395–1405.
- [23] K. Wang, X. Quan, and R. Wang, "BiSET: Bi-directional selective encoding with template for abstractive summarization," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2153–2162.
- [24] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, "Neural abstractive text summarization with sequence-to-sequence models," *ACM Trans. Data Sci.*, vol. 2, no. 1, pp. 1–37, 2021.
- [25] W. Wu, W. Li, X. Xiao, J. Liu, Z. Cao, S. Li, H. Wu, and H. Wang, "BASS: Boosting abstractive summarization with unified semantic graph," 2021, *arXiv:2105.12041*.
- [26] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–18.
- [27] Y. Liu and M. Lapata, "Hierarchical transformers for multi-document summarization," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5070–5081.
- [28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
- [29] H. Wang, Y. Gao, Y. Bai, M. Lapata, and H. Huang, "Exploring explainable selection to control abstractive summarization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 13933–13941.
- [30] P. Cui and L. Hu, "Topic-guided abstractive multi-document summarization," in *Proc. Findings Assoc. Comput. Linguistics*, 2021, pp. 1463–1472.
- [31] L. Dong, M. N. Satpute, W. Wu, and D.-Z. Du, "Two-phase multidocument summarization through content-attention-based subtopic detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 6, pp. 1379–1392, 2021.
- [32] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. 4th Int. Conf. Learn. Represent.*, San Juan, Puerto Rico, Y. Bengio and Y. LeCun, Eds. May 2016, pp. 1–14.
- [33] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 1–8.
- [34] D. Antognini and B. Faltings, "Learning to create sentence semantic relation graphs for multi-document summarization," in *Proc. EMNLP-IJCNLP*, 2019, p. 32.
- [35] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2006, pp. 1735–1742.
- [36] K. Hong and A. Nenkova, "Improving the estimation of word importance for news multi-document summarization," in *Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2014, pp. 712–721.
- [37] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.
- [38] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.
- [39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
- [40] P. Diederik Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–15.
- [41] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "OpenNMT: Open-source toolkit for neural machine translation," *Proc. ACL*, Vancouver, BC, Canada, Jul. 2017, pp. 67–72.
- [42] Y. Wang, L. Wang, Y. Li, D. He, W. Chen, and T.-Y. Liu, "A theoretical analysis of NDCG ranking measures," in *Proc. 26th Annu. Conf. Learn. Theory*, vol. 8, 2013, p. 6.
- [43] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization Branches Out*, 2004, pp. 74–81.
- [44] J. G. Carbonell and J. Goldstein, "The use of mmr and diversity-based reranking for reordering documents and producing summaries. (1998)," *Citado*, vol. 4, pp. 24–42, Oct. 1998.
- [45] Y. Mao, Y. Qu, Y. Xie, X. Ren, and J. Han, "Multi-document summarization with maximal marginal relevance-guided reinforcement learning," 2020, *arXiv:2010.00117*.
- [46] S. I. K and S. R. Balasundaram, "Phrase embedding based multi document summarization with reduced redundancy using maximal marginal relevance," in *Proc. Int. Conf. Electr. Eng. Informat. (ICELTICs)*, Oct. 2020, pp. 1–5.



**YUN-CHIEN TSENG** received the B.S. and M.S. degrees in mathematics from National Taiwan Normal University (NTNU), in 2013 and 2015, respectively. She is currently pursuing the Ph.D. degree in computer science and engineering with National Yang Ming Chiao Tung University (NYCU). She is a Teaching Assistant with both NCTU and NTNU, where she assists in algebra, graph theory, and number theory courses. She is also a Teaching Assistant with the Houston Association of Space Science and Engineering, supporting a program for teenagers. Besides teaching experience, she has also been a Research Assistant in mining and conducting statistical analysis in student educating related data with the Shida Institute of Mathematics Education. Her current research interests include data mining, graph embedding, and machine learning.



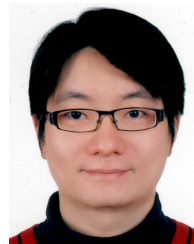
**MU-HUA YANG** received the M.S. degree in computer science and engineering from National Yang Ming Chiao Tung University, Taiwan, in 2021. She is currently an Engineer at Phison Electronic Corporation.



**WEN-CHIH PENG** (Member, IEEE) received the B.S. and M.S. degrees from National Yang Ming Chiao Tung University, Taiwan, in 1995 and 1997, respectively, and the Ph.D. degree in electrical engineering from National Taiwan University, Taiwan, in 2001. He was mainly involved in projects related to mobile computing, data broadcasting, and network data management. Currently, he is a Professor with the Department of Computer Science, National Yang Ming Chiao Tung University. He has served as a PC member in several prestigious conferences, such as the IEEE International Conference on Data Engineering (ICDE), ACM International Conference on Knowledge Discovery and Data Mining (ACM KDD), IEEE International Conference on Data Mining (ICDM), and ACM International Conference on Information and Knowledge Management (ACM CIKM). His research interests include mobile data management and data mining.



**YAO-CHUNG FAN** received the Ph.D. degree in computer science from National Tsing Hua University, Taiwan. His research interests include natural language processing, social data management, and text mining. His Ph.D. dissertation received the 2011 TIEEE Doctoral Dissertation Award. He was funded by the Taiwan National Science Council Award for visiting the Pennsylvania State University. Please check the Demo from his research group <https://demo.nlpnchu.org/>



**CHIH-CHIEH HUNG** (Member, IEEE) received the M.S. and Ph.D. degrees from the National Yang Ming Chiao Tung University, Taiwan, in 2005 and 2011, respectively. Currently, he is an Assistant Professor with the Department of Management Information System, National Chung Hsing University, Taiwan. He has published some papers in several prestigious conferences, such as IEEE International Conference on Data Engineering (ICDE), IEEE International Conference on Data Mining (ICDM), and ACM Conference on Information and Knowledge Management (ACM CIKM) and prestigious journals (e.g., IEEE TKDE, IEEE SMC, and VLDB J.). His research interests include data mining, mobile and pervasive computing, big data analytics, and artificial intelligence. He was a recipient of the Best Paper Award in ACM Workshop on location-based social network 2009.

• • •