Template-Based Headline Generator for Multiple Documents

Yun-Chien Tseng\(^1\), Mu-Hua Yang\(^1\), Yao-Chung Fan\(^2\), Wen-Chih Peng\(^1\) (Member, IEEE), and Chih-Chieh Hung \(^3\) (Member, IEEE)

\(^1\)Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
\(^2\)Department of Computer Science and Engineering, National Chung Hsing University, Taichung, Taiwan
\(^3\)Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan

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

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 abbreviation 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.

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.

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 embed-
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. |

- **Highlights** *** 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.
- **What You Can Expect** *** 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.***
- **Package Info** *** 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) ***
- **Important Info** *** 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 & 79th St, New York, NY 10024 ***
- **Additional Info** *** 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. ***
- **How to Redeem Your Voucher** *** Show your KKday e-voucher at ticketing counter for ticket exchange.

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.¹
- Experimental result shows our sentence selection method is able to choose key sentences that can keep relevance 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.

1Our dataset and code are available on: https://github.com/klks0304/mud2h
The method BASS connect inputs can connect words be-
applies semantic graph to connect words in input documents.

Recent publish method BASS [25] methods and conduct on CNN/Daily Mail dataset. Usually, seq2seq model, therefore a toolkit NATS [24] collect these subjects. Most of the abstraction-based summary method are representation, and assists machine generating user-specific
poses a personalized subject generation model, which adds a
expression for output, BiSET uses a bidirectional selective layer
method, follows the previous architecture. To improve ex-
the state-of-the-art template-based abstractive summarization
soft template-based architecture, which uses existing sum-
3
great deal of manual effort. Re
ally defined, it is very time-consuming and also requires a
be filled with the keywords. Because templates are manu-
template-based approaches [1] [22], a template using the
template-based summarization Re
proposed a novel
soft template-based architecture, which uses existing sum-
aries 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 ex-
pression for output, BiSET uses a bidirectional selective layer
with two gates to select key information. TempPES [3] pro-
poses 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 doc-
ument summarization. Recent publish method BASS [25]
Applies semantic graph to connect words in input documents.
The method BASS connect inputs can connect words be-
motivated abstraction. 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. Hier-
Summ [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 effi-
ciently 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 gen-
erating a representative headline for input documents, we
extract sentences to generate an overall meaning for the doc-
uments. 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 \( D = \{ d_i | i \in [1, N] \} \), where \( N \) is the number of documents. Each document \( d_i \) consists of a set of sentences \( S = \{ s_{i,j} | j \in [1, M_i] \} \), 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
tween 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. Hier-
Summ [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 effi-
ciently 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.
embedding as an input node feature. Applying sentence relation graph and sentence embedding on Graph Convolutional Network can produce a high-level hidden feature for each sentence. After that, we use a linear layer to estimate a salience score for each sentence. Giving salience score for each sentence helps model extract suitable sentence from documents. Finally, instead of selecting top salience score sentences, we use a greedy method to select the salience 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 salience 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 collection of documents are concatenated as the following:

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

   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 salience 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, $a_{ij} = 0$.

- $X \in \mathbb{R}^{M \times d_x}$, the input node feature matrix, where $d_x$ 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 $\tilde{A} = A + I_M$, where $I_M$ is the identity matrix. Our propagation rule follows:

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

$\tilde{A} = D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}}$ 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 $A$. $W_1$ is an input-to-hidden weight matrix to learn in the $l^{th}$ layer, and $b_i$ 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:

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

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

4) Training

Previous works [34] [13] 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 for ranking. To overcome this problem, we trained the model to output scores close to 0 or 1 which may cause an obstacle. Therefore, 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 [36] [12]. 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 $t$, the sentence is considered redundant. If not, we select the sentence. We repeat this step until the expected number of sentences $n$ is reached.

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 [36] [12]. 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 $t$, the sentence is considered redundant. If not, we select the sentence. We repeat this step until the expected number of sentences $n$ is reached. The algorithmic form us shown in Algorithm 2.

### Algorithm 2: Sentence Selection Algorithm

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

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

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

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

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

for $s \in S'$ do

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

Drop $s$;

end

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

Add $s$ to $C$;

end

if $|C| > n$ then

return $C$

end

return $C$

$R(s) = \text{softmax}(r(s))$, 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.

### 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 concatenated, hence it can be considered as an article. Each article \( A_r \) consists of \( n \) words \( \{ x^a_i \mid i \in [1,n] \} \). 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 \( T' = (x^a_1, x^a_2, \ldots, x^a_n) \) 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\(^2\) 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 \( h^t_n \) and \( h^t_1 \) 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 \( h^t_n \) and backward hidden state \( h^t_1 \) as the template representation \( h^t \). For each time step \( i \), it takes \( h^t \) and \( h^t_i \) as inputs to output a template gate vector \( g_i \) to select from \( h^t_i \):

\[
g_i = \sigma(W_a h^t_i + W_t h^t_i + b_a)
\]

\[
h^t_i = h^t \odot g_i
\]

where \( \sigma \) denotes the sigmoid activation function, and \( \odot \) is element-wise multiplication. After the T2A gate, we obtain a sequence of vectors \( (h^t_1, h^t_2, \ldots, h^t_n) \).

The goal of the A2T gate is to control the proportion of \( h^t \) 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 \( h^t_i \):

\[
d = \sigma((h^a)^TW_a h^t_i + b_d)
\]

\( h^a \) is generated in the same way as \( h^t \): concatenating the forward hidden state \( h^t_n \) and backward hidden state \( h^t_1 \).

The final article representation is computed by the weighted sum of \( h^t_i \) and \( h^a \):

\[
z^a_i = d \cdot (h^t_i) + (1 - d) \cdot (h^a_i)
\]

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 \( h^t_{t-1} \) generated in the previous step, and then outputs a new hidden state for the current step:

\[
h^t_t = LSTM(w_{t-1}, h^t_{t-1})
\]

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 \( h^t_t \) and \( z^a \) to get importance scores. The importance scores are then normalized to get the current context vector by weighted sum:

\[
c_t = \sum_{i=1}^{L} a_{t,i} z^a_i
\]

\[
a_{t,i} = \frac{\exp(c_{t,i})}{\sum_{i=1}^{L} \exp(c_{t,i})}
\]
 \[ e_{t,i} = (z^{c}_{t})^{T}W_{e,h_{t}^{c}} \]  

(14)

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 \( h_{t}^{r} \):

\[
h_{t}^{r} = \tanh \left(W_{a}[c_{t}; h_{t}^{c}] \right)
\]

(15)

In the final stage, \( h_{t}^{r} \) is fed into a softmax layer to output the target word distribution for predicting the next word \( w_{t} \) over existing words \( w_{1}, w_{2}, \ldots, w_{t-1} \):

\[
p(w_{t} | w_{1}, \ldots, w_{t-1}) = \text{softmax}(W_{p}h_{t}^{r})
\]

(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 \( w^{*} \):

\[
\mathcal{L}_{h} = - \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{L} \log p(w_{j}^{(i)} | w_{j-1}^{(i)}, x^{a(i)}, x^{t(i)})
\]

(17)

To learn the style of the template, we minimize the negative log-likelihood between generated headline \( w \) and the template \( w^{t} \):

\[
\mathcal{L}_{t} = - \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{L} \log p(w_{j}^{t(i)} | w_{j-1}^{t(i)}, x^{a(i)}, x^{t(i)})
\]

(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:

\[
\mathcal{L} = \mathcal{L}_{h} + \alpha \mathcal{L}_{t}
\]

(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 extractor?

A. DATASETS

We used a real-world dataset provided by the traveling e-commerce platform, KKday\(^3\). 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 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\(^4\). Overview of our dataset is describe in Table 2.

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

| Description       | Document with headline | Sentences |
|-------------------|------------------------|-----------|
| Blog              | 1813                   | 87377     |
| Introduction      | 31431                  | 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 epoch 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

\(^3\)https://www.kkday.com/zh-tw

\(^4\)https://github.com/klks0304/mud2h
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 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.

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 Relevence (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 Rough-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 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 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.

| method                  | example                                                                 |
|-------------------------|-------------------------------------------------------------------------|
| Top-k                   | ・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) |
|                         | ・日本大阪環球影城™標準7飛天翼龍7逆轉世界@快速60通關券KKday官網同步開賣(Uniform Express Pass 7: Standard, The Flying Dinosaur, or Backdrop) |
|                         | ・VIP手環提早入園享受專屬通關(Early park enter,enjoying exclusive pass with VIP wristband) |
|                         | ・省下多次轉車時間搭乘由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) |
|                         | ・立即Kday預訂大阪環球影城VIP手環門票在阿倍野16樓售票窗口領取直接入場立即Kday to book the Universal Studio Japan VIP wristband ticket, take your ticket and enter at Abeno 16F ticket counter) |
|                         | ・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) |
|                         | ・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事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.) |

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% 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.

| Model                  | ROUGE-1 | ROUGE-2 | ROUGE-L |
|------------------------|---------|---------|---------|
| Random                 | 31.26   | 11.70   | 21.99   |
| Top-k                  | 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   |
| SemSemSum [34]         | 40.62   | 15.64   | 27.22   |
| Ours(extractor)        | 42.62   | 16.50   | 27.89   |

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

| Model                  | Similarity |
|------------------------|------------|
| TextRank [7]           | 0.7498 ±0.0577 |
| Cont. LexRank [8]      | 0.7322 ±0.0586 |
| MMR [46]               | 0.7517 ±0.0568 |
| SemSemSum [34]         | 0.7302 ±0.0505 |
| Ours(extractor)        | 0.7533 ±0.0571 |

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

| method                  | example                                                                 |
|-------------------------|-------------------------------------------------------------------------|
| ours                    | ・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) |
|                         | ・日本大阪環球影城™標準7飛天翼龍7逆轉世界@快速60通關券KKday官網同步開賣(Uniform Express Pass 7: Standard, The Flying Dinosaur, or Backdrop) |
|                         | ・VIP手環提早入園享受專屬通關(Early park enter,enjoying exclusive pass with VIP wristband) |
|                         | ・省下多次轉車時間搭乘由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) |
|                         | ・立即Kday預訂大阪環球影城VIP手環門票在阿倍野16樓售票窗口領取直接入場立即Kday to book the Universal Studio Japan VIP wristband ticket, take your ticket and enter at Abeno 16F ticket counter) |
|                         | ・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) |
|                         | ・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事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.) |

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)
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     |
| MuD2H         | 19.05   | 16.96   | 0.53       |

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

| Sample | Sample 2 | Sample 3 | Average |
|--------|----------|----------|---------|
| MMR    | 2.839    | 2.742    | 2.839   | 3.075   |
| SemSentSum | 2.903    | 2.742    | 2.835   | 2.860   |
| MuD2H  | 2.419    | 2.096    | 2.483   | 2.353   |
| Human-written | 1.839    | 1.548    | 1.742   | 1.710   |

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 chick 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-calculations ensure 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

Thanks to KKDay’s data group for helping and providing dataset. Thanks to Yu-Chien Tang’s help in experiments.
Table 10. Result of the personalized headline generated by our full model.

| 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                      |
|----------------------------|
| 【宜蘭景點】朝濯獨木舟體驗 (Yilan Paradise: Chawowan Kayak Experience) |

| User3                      |
|----------------------------|
| 【宜蘭過去景點】福隆海濱公園+獨木舟體驗 (Must visit attractions in Yilan: Fulong Ticket + Canoe experience) |

References

[1] Liang Zhou and Eduard Hovy. Template-filtered headline summarization. In Text summarization branches out, pages 56–60, 2004.

[2] Tianshang Liu, Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Huang. Heterogeneous graph neural networks for extractive document representations (ICLR), 2017.

[3] Diego Antognini and Boi Faltings. Learning to create sentence semantic representations (ICLR), 2018.

[4] Peng Cui and Le Hu. Topic-guided abstractive multi-document summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1095–1104, 2013.

[5] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[6] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, and Kyunghyun Cho. Abstractive text summarization using sequence-to-sequence models. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 379–389, 2015.

[7] Ruipeng Jia, Yanan Cao, Haihao Shi, Fang Fang, Pengfei Yin, and Shiyang Wang. Flexible non-autoregressive extractive summarization with threshold: How to extract a non-fixed number of summary sentences. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 13134–13142, 2021.

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2, pages 3104–3112, 2014.

[9] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 379–389, 2015.

[10] Alex Ihler, Dipo Nwogu, and Jared Seiler. Exploring content models for focused meeting summarization. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1395–1405, 2013.

[11] Kai Wang, Xiaojun Quan, and Rui Wang. Biset: Bi-directional selective encoding with template for extractive summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2153–2162, 2019.

[12] Tiansheng Liu, Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Huang. Heterogeneous graph neural networks for extractive document representations (ICLR), 2017.

[13] Lu Wang and Claire Cardie. Domain-independent abstract generation for focused meeting summarization. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1395–1405, 2013.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakub Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in Neural Information Processing Systems, 30:5998–6008, 2017.

[15] Lu Wang and Claire Cardie. Domain-independent abstract generation for focused meeting summarization. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1395–1405, 2013.

[16] Peng Cui and Le Hu. Topic-guided abstractive multi-document summarization. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1463–1472, 2021.

[17] Yuqiang Dong, Menghna N Satpute, Weili Wu, and Ding-Zhu Du. Two-phase multidocument summarization through content-attention-based subtopic detection. IEEE Transactions on Computational Social Systems, 8(6):1379–1392, 2021.

[18] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In Yoshua Bengio and Yann LeCun, editors, 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings, 2016.

[19] Arash Renshaw, Alireza Rabbani, and Stephen Soderland. Towards learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), volume 2, pages 1735–1742. IEEE, 2006.

[20] Kai Hong and Ani Nenkova. Improving the estimation of word importance for news multi-document summarization. In Proceedings of the 14th
YAO-CHUNG FAN received the Ph.D. degree in computer science from National Tsing Hua University, Taiwan. Dr. Fan’s Ph.D. dissertation received the 2011 IEEE Doctoral Dissertation Award. Dr. Fan was funded by Taiwan National Science Council Award for visiting the Pennsylvania State University. His research interests are natural language processing, social data management, and text mining. Please check the Demo from Dr. Fan’s research group https://demo.nlpnchu.org/

WEN-CHIH PENG received the BS and MS degrees from National Chiao Tung University, Taiwan, in 1995 and 1997, respectively, and the PhD degree in electrical engineering from National Taiwan University, Taiwan, in 2001. Currently, he is a professor in the Department of Computer Science, National Chiao Tung University, Taiwan. Prior to joining the Department of Computer Science, National Chiao Tung University, he was mainly involved in projects related to mobile computing, data broadcasting, and network data management. 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. He is a member of the IEEE.

MU-HUA YANG received her M.S. in Computer Science and Engineering from National Yang-Ming Chiao-Tung University (NYCU). She has been a teaching assistant in Houston Association of Space Science and Engineering, supporting a program for teenagers. Besides teaching experience, she has also been a research assistant in Shida Institute of Mathematics Education, mining and conducting statistical analysis in student educating related data. Her current research interests include data mining, graph embedding and machine learning.

CHIH-CHIEH HUNG received the MS and PhD degrees from the National Chiao Tung University, Taiwan, in 2005 and 2011, respectively. Currently, he is an assistant professor at the department of management information system, National Chung Hsing University, Taiwan. Dr. Hung 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, VLDB J.). Dr. Hung has the best paper award in ACM Workshop on location-based social network 2009. His research interests include data mining, mobile and pervasive computing, big data analytics, and artificial intelligence.

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/