Multidocument Aspect Classification for Aspect-Based Abstractive Summarization

Ye Wang, Yingmin Zhou, Mengzhu Wang, Zhenghan Chen, *Graduate Student Member, IEEE*, Zhiping Cai, Junyang Chen, *Member, IEEE*, and Victor C. M. Leung, *Life Fellow, IEEE*

***Abstract*— Multidocument aspect-based summarization (AspSumm) aims to generate focused summaries based on the target aspects from a cluster of relevant documents. Generating such summaries can better satisfy readers' specific points of interest, as readers may have different concerns about the same articles. However, previous methods usually generate aspect-based summaries based on the given aspects without using the relationship among aspects to assist in the summarization. In this work, we propose a two-stage general framework for multidocument AspSumm. The model first discovers the latent relationship among aspects and then uses relevant sentences selected by aspect discovery to generate abstractive summaries. We exploit latent dependencies among aspects using a tag mask training (TMT) strategy, which increases the interpretability of the model. In addition to improvements in summarization over aspect-based strong baselines, experimental results show that our proposed model can accurately discover multidomain aspects on the WikiAsp dataset.**

***Index Terms*— Aspect-based summarization (AspSumm), multidocument summarization, pretrained model.**

I. Introduction

RECENT years have seen remarkable methods in generating generic summaries for multiple documents [1], [2], [3], [4], [5], [6]. Multidocument summarization has a practical significance which covers multiple topics or aspects. In this study, we tackle multidocument aspect-based summarization (AspSumm), which aims to generate aspect-related summaries where each summary contains the appropriate aspect-related information. Such focused summaries corresponding to specific points of interest usually meet particular information needs. The task is traditionally decomposed into two stages: aspect discovery and AspSumm aiming to generate summaries in line with discovered aspects.

There are several methods for exploring the problem of aspect-based abstractive summarization [6], [7], [8]. However, existing AspSumm works usually focus specifically on the given aspects without considering their relationships. In fact, there are certain relationships among aspects, for example, *census* is likely to co-occur with *demographics* in the *Town* domain, while *prelude* is unlikely to co-occur with *background* in the *Event* domain. Latent dependencies in different domains can help more accurate classification, which in turn helps better summarization. In addition, most of the works on aspect discovery need to manually set parameters [9]. As a result, models trained on different domain data need to select a suitable threshold through multiple attempts.

In this work, we propose a two-stage general framework consisting of an aspect classifier using a tag mask training (TMT) process and a summary generator. Inspired by pretrained models, which have been employed as encoders for detecting entailment relationships [10], we expand aspect embeddings with tag embeddings into the model to study the latent relationship among various aspects. The tag embeddings indicate the states of aspects as *true*, *false*, or *unknown*. We demonstrate that the aspect discovery method achieves superior results on different domains benefiting from the TMT strategy. The strategy predicts the states of aspects masked by setting the tag embeddings as *unknown*. The summary generator relies on the sharing embeddings learned from the aspect classifier to generate abstractive summaries. Moreover, explicitly inducing latent dependencies can increase the interpretability of the model during training and inference by setting tag embeddings to *true* or *false*. When the number of aspects changes, the aspect information can be easily modified by adjusting aspect embedding layers. Such a framework has better generalization ability for multidocument aspect-based abstractive summarization and it is suitable for various domains.

We evaluate the proposed model on the recently released WikiAsp dataset, as WikiAsp has explicit aspect labels. Our model brings more aspect-relevant and meaningful summaries compared to aspect-based baselines. Besides, there are substantial improvements over the strong baseline for aspect discovery. We investigate the influence of latent dependencies among aspects, and the ablation study shows that inducing

Manuscript received 17 October 2022; revised 28 January 2023; accepted 22 February 2023. Date of publication 13 March 2023; date of current version 31 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62072465, in part by the Science and Technology Innovation Program of Hunan Province under Grant 2022RC3061, in part by the National Natural Science Foundation of China under Grant 62102265, in part by the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy [Shenzhen (SZ)] under Grant GML-KF-22-29, and in part by the Natural Science Foundation of Guangdong Province of China under Grant 2022A1515011474. *(Corresponding authors: Zhiping Cai; Junyang Chen.)*

Ye Wang, Mengzhu Wang, and Zhiping Cai are with the College of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: wangye19@nudt.edu.cn; dreamkily@gmail.com; zpcai@nudt.edu.cn).

Yingmin Zhou is with Baidu Inc., Beijing 100080, China (e-mail: zhouyingmin@baidu.com).

Zhenghan Chen is with the School of Software and Microelectronics, Peking University, Beijing 100091, China (e-mail: 1979282882@pku.edu.cn).

Junyang Chen and Victor C. M. Leung are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518061, China (e-mail: junyangchen@szu.edu.cn; vleung@ieee.org).

Digital Object Identifier 10.1109/TCSS.2023.3252723

latent dependencies helps aspect classification and high-quality summaries generation.

In summary, the benefits of our proposed model are as follows.

1) We explore latent dependencies among aspects by a novel dynamic TMT process and propose a two-stage general training framework that can generate more focused summaries relevant to target aspects.
2) Our approach can unify the aspect discovery and abstractive summarization into one architecture through the learned sharing embeddings and substantially promote the classification of aspects in all 20 different domains.
3) In addition to the promotion of aspect discovery, experiments on the recently released multidocument AspSumm dataset, WikiAsp, show that our model achieves state-of-the-art performance.

II. Related Work

AspSumm has been widely investigated primarily in the customer feedback domain [11], [12], [13], [14], which extracts sentiment and information according to product properties or restaurant reviews. Angelidis and Lapata [15] proposed a weakly supervised method, which can discover aspects through topic models and did not require gold aspect annotations. Bražinskas et al. [16] defined a generative model that used the unsupervised setting to capture the intuition by a hierarchical variational autoencoder model for controlling the novelty of the new review. Amplayo and Lapata [17] introduced several linguistically motivated noise generation functions and a summarization model which learned to denoise the input and generated the original review which brought substantial improvements over both abstractive and extractive baselines. Mukherjee et al. [18] proposed an unsupervised approach to extract coherent aspects from tourist reviews and then proposed an integer linear programming (ILP)-based extractive technique to select an informative subset of opinions around the identified aspects while respecting the user-specified values for various control parameters. Recently, there has been plenty of work devoted to solving AspSumm in the newswire domain. Krishna and Srinivasan [7] utilized online encyclopedia entries and sections in multiple cited references for generating a lead section of a Wikipedia article. Frermann and Klementiev [6] compared models for AspSumm incorporating different aspect-driven attention mechanisms and generated news summarization to solve the lack of large-scale data with aspect annotation. Hayashi et al. [9] also built a large-scale dataset for multidomain AspSumm based on Wikipedia with different aspect annotations. Compared with our proposed method, previous methods ignore the latent dependencies among aspects.

On the other hand, query-based summarization aims to generate summaries corresponding to a natural language input query [19], [20], [21]. Query-based summarization was part of a shared task in 2005 with 32 participating automatic summarization systems [22]. Cao et al. [23] proposed a joined attention model AttSum to meet the query need and calculate sentence weight. Egonmwan et al. [24] built a two-step process for query-based abstractive summaries. Furthermore, previous studies on English Wikipedia generation [25], [26], [27] can be regarded as a multidocument summarization task, attempting to generate the lead sections of Wikipedia articles, but they do not generate different aspect-based summaries. In contrast, AspSumm tends to generate related summaries concerning one or a few words.

There have been substantial methods built on neural encoder–decoder architectures with attention [1], [2], [28]. Extractive models have achieved promising results on single-document summarization, but abstractive methods have increased attention, and some works adjusted the model to adapt to the multidocument summarization task. Recently, Transformer-based models have been utilized to solve the task of summarization [29], [30], [31]. Wang et al. [32] rearranged and explored the semantics learned by a topic model and proposed a topic assistant based on Transformer for abstractive summarization. Hayashi et al. [9] finetuned the RoBERTa-based classification model to obtain probabilities of each aspect for a given sentence and generated summaries based on a chunked paragraph that discussed the same aspect. Different from another large-scale multidocument [25] proposed for Wikipedia lead section generation, our model tends to generate AspSumm based on Wikipedia subtopics [9]. We regard the multidocument AspSumm as a two-stage process but interrelated: aspect discovery and AspSumm. The model discovers aspects through a dynamic TMT strategy and utilizes the sharing embeddings learned in the aspect discovery section to explicitly drive summary generation. Sentence selection based on ranked attention scores can lead to generating more relevant summaries of the target aspect.

III. Method

The multidocument AspSumm task aims to generate focused summaries relevant to the aspects. Following [9], we regard the multidocument AspSumm as a two-stage task: aspect discovery and AspSumm. However, we make more use of the relationship between the two modules. Our model consists of an aspect classifier for aspect discovery and a summary generator for AspSumm. More specifically, we consider aspect discovery as a multilabel classification task that labels aspects related to the input of multiple documents. The aspect classifier reads multidocument features extracted by the Longformer model [33] and learns the latent dependencies among aspects through the dynamic TMT method. The correctly classified aspects will guide the input of the aspect-based summary generation by explicitly ranking multihead attention scores. Besides, the learned sharing embeddings induce the latent dependencies into the summary generator. Sections III-A–III-A2 describe the aspect discovery method, especially the dynamic TMT process and the Transformer-based summary generator with induced latent dependencies. Fig. 1 shows the overview of our approach.

A. Aspect Discovery

1) Modeling Multidocuments and Aspects:

a) Sentence embeddings $\mathbf{S}$*:* Given a cluster of documents, we encode the whole multidocument using the Longformer encoder [33] that initializes parameters from

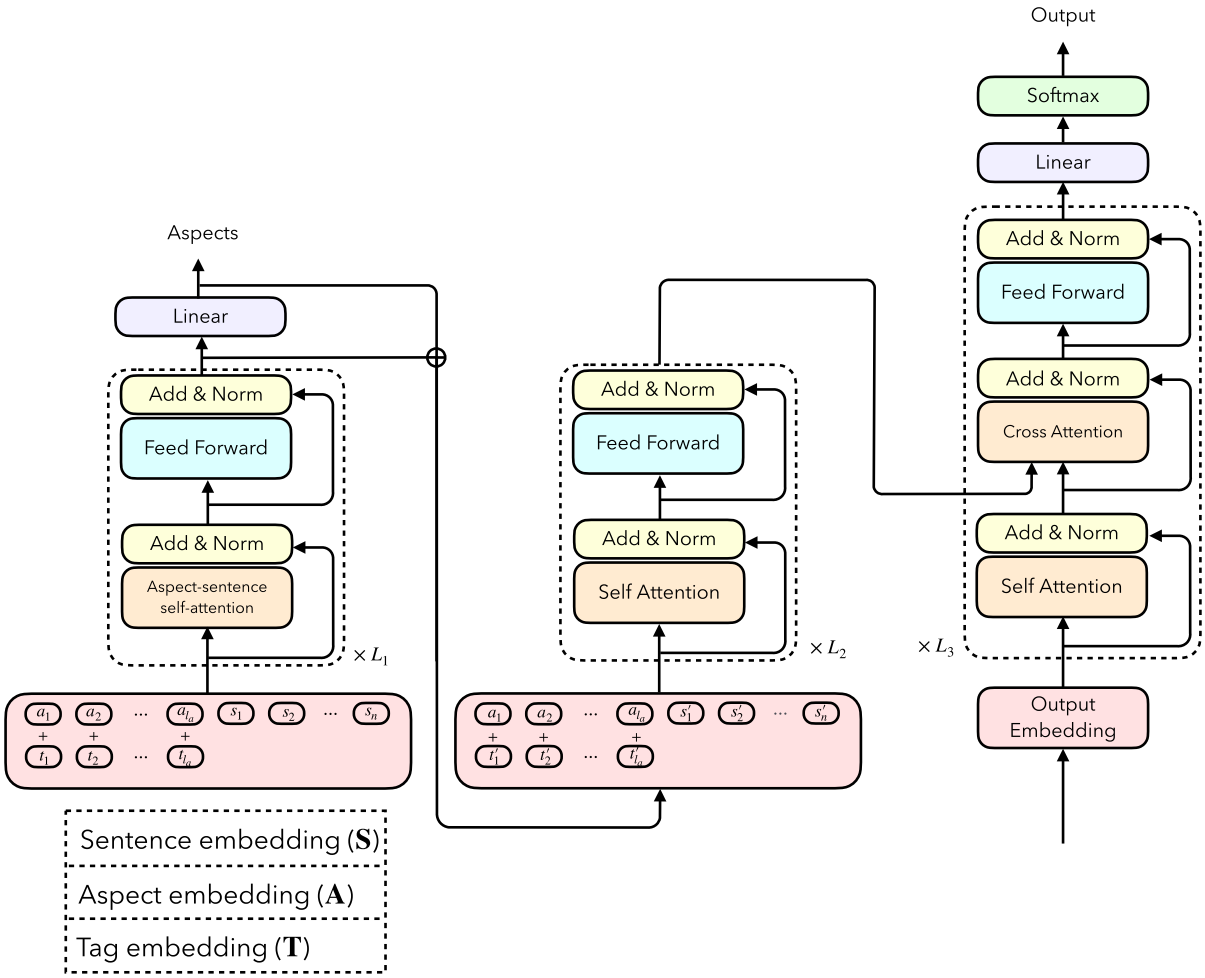


Fig. 1. Illustration of our approach. Left: Aspect discovery. Middle: AspSumm encoder. Right: AspSumm decoder.

RoBERTa [34] and is fine-tuned during the training. Following [35], we, respectively, insert <s> and </s> tokens at the beginning and the end of each sentence without adding global tokens. The input multidocument after tokenization is denoted as $D = \{s_1, \ldots, s_n\}$ and $s_i = \{w_{i1}, \ldots, w_{il_i}\}$, where l_i is the number of BPE tokens in the ith sentence including special tokens, that is, <s> token and </s> token. The Longformer model is then used to encode the multidocument

$$\{\mathbf{w}_{11}, \ldots, \mathbf{w}_{nl_n}\} = \text{LongFormer}(\{w_{11}, \ldots, w_{nl_n}\}) \quad (1)$$

where $\{\mathbf{w}_{11}, \ldots, \mathbf{w}_{nl_n}\}$ is the Longformer encoder output of the multidocument.

After the Longformer encoder, the <s> token representation can be used as sentence representation $\mathbf{S}$. We reconstruct sentence embeddings $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ by extracting the <s> token representation and considering the sentence representation $\mathbf{S}$ as text features, which will be sent to the classification model.

b) Aspect embeddings $\mathbf{A}$*:* For every domain with predefined set of aspects, $A = \{a_1, \ldots, a_{l_a}\}$. As aspects may be formed by a group of words, we retrieve aspect embeddings from a trainable embedding layer of size $l_a \times d_{\text{model}}$ and train from scratch, where l_a is the number of aspects and d_{model} is the embedding dim.

c) Tag embeddings $\mathbf{T}$*:* Inspired by the positional embedding proposed in Transformer [29] that incorporates explicit relative position dependency, we propose a method to incorporate aspect state information into aspects. Our proposed method explicitly indicates the aspect state as *true*, *false*, and *unknown*, which can be easily set in the training and inference process. During training, if the tag is set to *true*, it means that this multidocument contains the aspect and vice versa. On the other hand, when we set the tag to *unknown*, the TMT process described in the next section will obtain the latent aspect dependencies. We denote aspect state embeddings $\tilde{a}$ by simply adding a tag embedding vector $\mathbf{t}_i$ over aspect embedding vector $\mathbf{a}_i$

$$\tilde{\mathbf{a}}_i = \mathbf{a}_i + \mathbf{t}_i \quad (2)$$

where $\mathbf{t}_i$ takes on one of three possible tags: *true*, *false*, and *unknown*. Similar to the setting of aspect embeddings, the tag embeddings are retrieved from a trainable embedding layer of size $3 \times d_{\text{model}}$.

We employ the Transformer encoder [29] to encode the aspects and sentences. The Transformer encoder of aspect discovery is a stack of L_1 identical sublayers. The input embeddings of the Transformer encoder can be obtained by simply concatenating the aspects state embeddings and

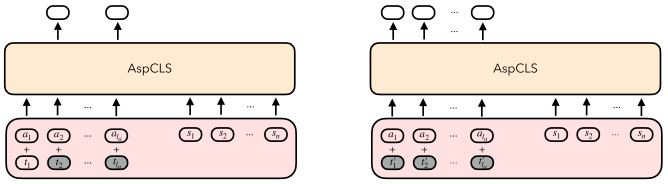


Fig. 2. Dynamic TMT strategy. Left: During training, the model predicts the dynamically masked input aspects. Right: During inference, the model predicts the classification of all aspects or a combination of known aspect inputs.

sentence embeddings $\mathbf{Z} = \{\tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_{l_a}, \mathbf{s}_1, \ldots, \mathbf{s}_n\}$. Specially, we put the aspect state embeddings at the left because documents are of variable length, and always padding at the right. The pad between aspect state embeddings and sentence embeddings will result in semantic incoherence.

We perform multihead self-attention to get the latent interactions between aspects and sentences, which combines knowledge of the same attention pooling via different representation subspaces of queries, keys, and values. In Transformers, the scaled dot-product attention is applied

$$\text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} \tag{3}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent query, key, and value matrices, respectively.

The input dimension of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the same and the attention is performed multiple times. The output heads are concatenated as the final output hidden state $\mathbf{h}$. The output of multihead attention is denoted as

$$\text{head}_i = \text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \tag{4}$$

$$\mathbf{h} = \text{concat}([\text{head}_1, \ldots, \text{head}_n]). \tag{5}$$

We update each layer output following a position-wise feed-forward network (FFN) with the residual connection and layer normalization:

$$\mathbf{Z}' = \text{layernorm}(\mathbf{h} + \text{FFN}(\mathbf{h})) \tag{6}$$

and stack the layers sequentially to form a Transformer encoder. We denote the final output of the Transformer encoder after multiple layers as $\mathbf{Z}' = \{\tilde{\mathbf{a}}'_1, \ldots, \tilde{\mathbf{a}}'_{l_a}, \mathbf{s}'_1, \ldots, \mathbf{s}'_n\}$.

2) Dynamic TMT: Adding tag embeddings (2) can easily incorporate aspect state as input to the Transformer encoder. Inspired by RoBERTa's "dynamic masking training" process [34], we introduce a novel dynamic TMT procedure with a Transformer encoder. It forces the model to learn latent correlations among aspects and allows the model suitable for any inference setting. The dynamic mask training method generates different masking patterns every time allowing the model to contact different versions of the same sentence and learns more dependencies. Fig. 2 gives an overview of this strategy.

We dynamically mask certain aspects during training by adding *unknown* tag embedding with other ground-truth aspects to predict the masked ones. Unlike masked language model training by masking random tokens, we dynamically mask a subset of aspects from a predefined set of inputs.

Most masked language model training methods [10], [34] mask out around 15% of the words. According to the ratio between aspects and sentences, we dynamically mask at least 25% of the aspects. That is, given l_a predefined aspects, there are a number of *unknown* aspects. The number of *unknown* aspects n is chosen from $0.25l_a$ to l_a randomly. n unknown aspects are sampled dynamically from all predefined aspects. Concatenate the rest known aspects that do not need to be predicted plus the corresponding ground-truth tag embeddings to predict the unknown aspects.

Through dynamically masking a random number of aspects during training, the model can learn many possible aspect combinations which help the model learn latent dependencies among aspects. In addition, it is suitable for training and inference procedures to expand any number of aspects or add any known information.

After modeling text features and aspects via the conditional pretrained encoder, we exploit a classifier to obtain the final aspect predictions, which is denoted as *AspCLS*. We use an independent FFN for the final aspect embedding $\tilde{\mathbf{a}}'_i$, which contains a single linear layer

$$\hat{y} = \text{FFN}(\tilde{\mathbf{a}}') = \sigma\left(\left(\mathbf{W}_c \cdot \tilde{\mathbf{a}}'\right) + b_c\right) \tag{7}$$

where σ is a sigmoid function, $\mathbf{W}_c \in \mathbb{R}^{d_{\text{model}} \times d_{\text{aspect}}}$ is the aspect weight, $b_c \in \mathbb{R}^{d_{\text{aspect}}}$ is the bias vector, and d_{aspect} is the number of aspects.

Following [36], our TMT pipeline tries to minimize the binary cross-entropy loss

$$L_{cls} = \frac{1}{N}\sum_{n=1}^{N} \text{BCE}\left(\hat{y}_u^{(n)}, y_u^{(n)} | y_k\right) \tag{8}$$

where $\text{BCE}(\cdot|y_k)$ represents the binary cross-entropy loss of *unknown* aspects. $\hat{y}_u$ denotes the predicted aspect states, y_u denotes the ground-truth masked aspect states, and y_k denotes the unmask aspect states. N denotes the number of samples in the corpus.

B. Aspect-Based Summarization

1) Sentence Selection: Fig. 3 shows an overview of the sentence selection method. As the second stage of multidocument AspSumm, it is essential for choosing aspect-related sentences. Benefiting from the dynamic TMT process, the model can be used for document segmentation with aspect-driven attention. We choose aspect-related sentences based on attention scores calculated in aspect discovery. The multihead attention scores in the last encoder layer represent correlations between aspects and sentences. Aspect-related sentences will guide a generation of higher-quality aspect-based summaries. By simply attentive adding aspect-to-sentence and transposed sentence-to-aspect multihead attention matrices, we obtain the aspect-related sentences according to sorted attention scores

$$\alpha = \sigma([\text{att}_{a2\ s}; \text{att}_{s2a}]\mathbf{W}_{\text{att}} + b_{\text{att}}) \tag{9}$$

$$\text{S}' = \text{sorted}(\text{masked}(\alpha\text{att}_{a2\ s} + (1-\alpha)\text{att}_{s2a}))) \tag{10}$$

where σ is the sigmoid function, $\mathbf{W}_{\text{att}}$ and b_{att} are trainable parameters, and $\text{att}_{a2\ s}$ and att_{s2a} represent aspect-to-sentence

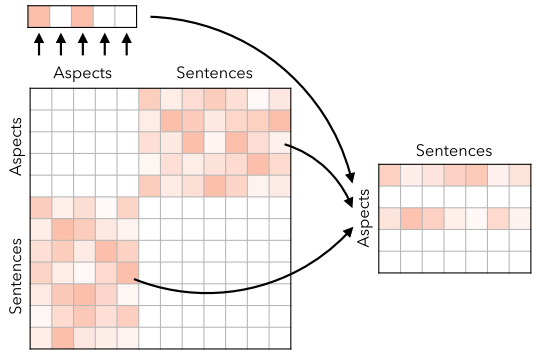


Fig. 3. Sentence selection. Select sentences related to correctly classified aspects by ranking attention scores.

and transposed sentence-to-aspect multihead attention matrices respectively. masked(·) represents attention matrices of correctly classified aspects.

Benefiting from the dynamic TMT process, the multihead attention matrices can represent the relationship between aspects and sentences. Otherwise, the model cannot be used for document segmentation without aspect-driven attention. And simply inducing aspect-driven attention by calculating attention between aspects and sentences cannot exploit latent structure among aspects [6].

2) Summary Generator: Aspect discovery model forms chunked paragraphs by self-attention that discuss the same aspects, denoted as $\mathbf{S}'_{\mathbf{a}}$, which become the input sequence to a summarization model. Besides, we concatenate the aspect state embedding $\tilde{\mathbf{a}}$ from the aspect discovery model. Adding the *true* tag embedding to each correctly classified aspect can easily induce aspect relationship into the summary generator. The summary generator is based on Transformer [29] which induces latent dependencies information to AspSumm by learned sharing aspect state embeddings.

The Transformer encoder and decoder of AspSumm consist of L_2 and L_3 stacks of identical sublayers. The Transformer encoder here is similar to aspect discovery's Transformer encoder; however, the sublayers in the Transformer decoder consist of three parts: a masked multihead self-attention mechanism, a multihead cross-attention mechanism, and a fully connected FFN. In addition, positional encodings are added to the input embeddings at the bottom of the decoder stacks. We denote the output of the lth layer as d_l and the input sequence for the first layer as d_0.

The self-attention sublayer is similar to the transformer encoder in the aspect discovery model. The output of the self-attention is fed to the cross-attention sublayer and FFN

$$d_l = \text{layernorm}(d_h + \text{FFN}(d_h)) \tag{11}$$

$$d_h = \text{layernorm}\big(\tilde{d} + \text{attention}\big(\tilde{d}, o, o\big)\big) \tag{12}$$

where o is the output of the Transformer encoder and $\tilde{d}$ is the encoder input.

The probability p_t of the next word over the target vocabulary is calculated by feeding the final output $d^t_{L_3}$ at the position t to a softmax layer

$$p_t = \text{softmax}\big(d^t_{L_3}\mathbf{W}_g + b_g\big) \tag{13}$$

TABLE I
LIST OF DOMAINS AND THE NUMBER OF WIKIPEDIA ARTICLES IN EACH DOMAIN

Domain	Train	Valid	Test
Album	24434	3104	3038
Animal	16540	2005	2007
Artist	26754	3194	3329
Building	20449	2607	2482
Company	24353	2946	3029
EducationalInstitution	17634	2141	2267
Event	6475	807	828
Film	32129	4014	3981
Group	11966	1462	1444
HistoricPlace	4919	601	600
Infrastructure	17226	1984	2091
MeanOfTransportation	9277	1215	1170
OfficeHolder	18177	2218	2333
Plant	6107	786	774
Single	14217	1734	1712
SoccerPlayer	17599	2150	2280
Software	13516	1637	1638
TelevisionShow	8717	1128	1072
Town	14818	1911	1831
WrittenWork	15065	1843	1931

TABLE II
MODEL PERPLEXITY (ANIMAL DOMAIN; VALIDATION SET) UNDER DIFFERENT COMBINATIONS OF ASPECT DISCOVERY AND ASPSUMM LEARNING RATES

l_s \ l_c	1e-2	1e-3	1e-4	1e-5
1e-2	43.89	35.14	23.72	20.83
1e-3	39.10	27.48	21.44	16.47
1e-4	36.02	22.76	13.23	**7.62**
1e-5	37.83	24.17	15.63	8.31

where $\mathbf{W}_g \in \mathbb{R}^{d_{\text{model}} \times d_{\text{vocab}}}$, $b_g \in \mathbb{R}^{d_{\text{vocab}}}$, and d_{vocab} is the size of target vocabulary.

At last, the goal is to maximize the probability of output summary. The following negative logarithm likelihood function is optimized:

$$L_{\text{abs}} = -\frac{1}{N_a}\sum_{n=1}^{N_a} \log p\big(y_w^{(n)}|s\big) \tag{14}$$

where y_w is the ground-truth summary and N_a is the number of samples in the summarization corpus.

IV. EXPERIMENT

A. Dataset

We experiment with the latest released WikiAsp dataset [9], which is a large-scale dataset for multidocument AspSumm. It consists of instances in 20 domains where each domain has ten predefined aspect classes. The number of Wikipedia articles in each domain is shown in Table I. Source documents are a cluster of related documents referenced by a Wikipedia article. The target length for every aspect in each domain varies exceedingly, paired with a document cluster containing more than 5000 words discussing different aspects. As the original multidocument input is too long without a boundary, we preprocess the dataset by filtering sentences under

TABLE III
ASPECT DISCOVERY RESULTS ON THE TEST SET

	RoBERTaCLS [9]			AttentionXML [43]			X-Transformer [44]			AspCLS		
	Prec	Rec	F-1	Prec	Rec	F-1	Prec	Rec	F-1	Prec	Rec	F-1
Album	19.64	86.43	30.64	23.42	75.96	35.80	26.17	69.21	37.98	59.00	38.59	**46.66**
Animal	34.69	84.08	45.52	38.15	78.33	51.31	40.26	74.38	52.24	65.83	52.06	**58.14**
Artist	26.32	75.24	36.72	29.00	59.27	38.94	30.97	56.74	40.07	45.13	39.71	**42.25**
Building	31.46	91.25	42.92	40.68	82.24	54.43	49.76	77.20	60.51	80.58	63.16	**70.81**
Company	28.97	91.50	41.06	32.33	82.69	46.49	36.85	76.28	49.69	84.56	69.50	**76.30**
Educational.	25.64	93.82	37.66	28.71	84.39	42.84	31.11	80.20	44.83	66.26	58.32	**62.03**
Event	28.99	96.44	42.36	35.04	80.27	48.78	36.42	76.34	49.31	70.47	78.32	**74.19**
Film	32.84	91.46	45.17	34.94	85.28	49.57	35.48	83.61	49.82	62.77	60.27	**61.49**
Group	17.46	95.56	28.18	29.44	82.21	43.35	31.29	81.95	45.29	61.15	60.42	**61.24**
Historic.	33.38	90.22	42.98	38.85	82.37	52.80	40.06	80.39	53.47	77.09	67.31	**71.87**
Infras.	28.38	94.00	41.00	35.33	86.14	50.11	35.96	85.20	50.57	76.80	83.21	**79.88**
MeanOf.	23.24	83.13	33.88	25.97	77.58	38.91	28.72	76.84	41.81	52.33	51.63	**51.98**
OfficeHolder	21.22	73.25	30.62	23.12	69.94	34.75	23.87	68.88	35.45	38.83	34.86	**36.74**
Plant	31.25	83.17	42.10	35.23	81.02	49.11	37.16	80.45	50.84	62.45	51.35	**56.36**
Single	25.36	88.33	37.16	27.73	85.49	41.88	29.53	83.94	43.69	64.47	46.60	**54.10**
SoccerPlayer	28.54	67.18	37.16	30.13	59.46	40.00	32.95	58.74	42.22	53.76	47.16	**50.24**
Software	31.52	94.65	45.10	37.46	86.32	52.25	40.31	83.44	54.36	64.60	79.13	**71.13**
Television.	20.44	81.76	31.28	19.97	81.54	32.08	20.56	80.42	32.75	41.41	29.48	**34.44**
Town	42.61	71.85	50.12	45.95	67.42	54.65	49.83	63.05	55.67	85.55	81.74	**83.60**
WrittenWork	21.50	94.29	33.71	24.62	88.32	38.51	25.47	87.12	39.42	54.16	55.38	**54.76**
AVG	27.67	86.38	38.77	31.80	78.81	45.32	34.14	76.22	47.16	63.36	57.46	**59.91**

the threshold of term frequency inverse document frequency (TF-IDF) similarity.

B. Implementation Details

1) Text Feature Extractor: We use the same settings and the pretrained feature extractor as the previous state-of-the-art in each domain for a fair comparison. For all domains, we use the Longformer pretrained on CNN-daily mail (CNN/DM) as the feature extractor. Since the output dimension of the Longformer base is 768, we set our embedding size d_{model} as 768.

2) Transformer Model: We set six heads for multihead self-attention, multihead cross-attention, and masked multihead self-attention. We set the number of conditional pretrained encoder layers L_1, summary generator encoder layers L_2, and summary generator decoder layers L_3 to 3, 6, and 6.

3) Optimization: The model is trained on one Tesla V100 with 32 GB memory. For training, Adam [37] is used as the optimizer with betas = (0.9, 0.999). We set different learning rate initialization and weight decay for aspect discovery and AspSumm. For aspect discovery, we set the learning rate l_c to $1e{-}5$ and weight decay to 0. Besides, the learning rate is halved if the validation metric $F1$ on the development decreases for two consecutive epochs. For AspSumm, we set the learning rate l_s to $1e{-}4$ and weight decay to $1e{-}5$. The validation metric for AspSumm is loss, and the learning rate is halved if loss increases for two consecutive epochs on the development set. As our model uses separate learning rates for aspect discovery and AspSumm, we examine whether the combination of different learning rates is indeed beneficial. However, as the training, valid, and test sets are provided by WikiAsp [9], it is not appropriate to use the cross-validation strategy [38], [39], [40]. Specifically, we report model perplexity on the "Animal" domain validation set for varying learning rates in Table II. We can see that the model performs best with $l_c = 1e-5$ and $l_s = 1e-4$. We train the model with a mini-batch size of 2. We use dropout $p = 0.1$ for regularization. We disallow the same trigram repeating [41], [42] and use beam search with a beam size of 5 for summary decoding.

C. Metrics and Baselines

Following previous work [9], we evaluate two axes of the model: aspect discovery and AspSumm. Besides, each domain is evaluated individually because the aspect sets differ in different domains.

1) Aspect Discovery: We use precision, recall, and $F1$ to evaluate the performance of aspect discovery.

We compare aspect discovery with some strong baseline proposed in the latest years [9], [43], [44]. The baseline model, denoted as RoBERTaCLS, is a pretrained RoBERTa model with a sigmoid function to obtain probabilities of each aspect for a given sentence. AttentionXML [43] uses BiLSTMs and label-aware attention as the scoring functioning and performs warm-up training of the models with hierarchical label trees. X-Transformer [44] is a scalable approach to fine-tuning deep transformer models for multilabel text classification.

2) Aspect-Based Summarization: We use ROUGE [45] to evaluate the produced summary in our experiments. Following previous work [5], we report ROUGE $F1$ on the WikiAsp dataset. We compare our model with several baselines including typical methods and models proposed in the latest years.

Lead-3 [6] is an extractive baseline that concatenates the first three sentences of each source document as a summary. The source sentences here are the sorted sentences calculated through aspect-sentence attention scores. TextRank [9] is a graph-based ranking model for extracting important sentences. For the abstractive model, pointer generator network (PGN) [3] is an encoder–decoder architecture. Source factor (SF) [6] treats the target aspect as additional information based on PGN [3]. PreSumm [35] is a Transformer-based

TABLE IV
ASPSUMM RESULTS ON THE TEST SET. THE RESULTS OF TextRank AND PreSumm$_1$ ARE FROM [9] WHICH USES THE CLASSIFICATION RESULTS OF ROBERTACLS. Lead-3, PGN, SF, PreSumm$_2$, AND Longformer, and Our Model AspSumm Uses the Classification Results of AspCLS

	Lead-3 [6]			TextRank [9]			PGN [3]			SF [6]			PreSumm$_1$ [9]			PreSumm$_2$ [35]			Longformer [33]			AspSumm			Extractive Oracle		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Album	20.84	3.97	18.62	19.56	2.81	17.26	17.77	2.02	15.06	24.35	5.92	22.16	22.76	6.31	20.27	25.03	7.33	23.17	23.31	6.47	21.06	**29.69**	**8.68**	**27.52**	37.72	12.58	33.19
Animal	19.97	4.83	17.25	18.00	3.16	16.05	15.83	2.15	13.21	26.94	7.08	24.77	27.11	8.08	25.01	28.73	8.72	26.35	27.35	8.17	25.83	**30.14**	**8.94**	**28.60**	34.82	10.52	31.01
Artist	17.36	3.29	15.17	17.22	2.49	15.58	14.96	1.58	12.33	23.35	4.37	20.06	21.79	3.76	20.00	23.59	5.13	21.71	22.06	3.93	20.88	**26.43**	**6.45**	**24.18**	41.49	15.04	37.64
Building	24.06	5.47	21.83	23.91	4.96	21.85	20.39	3.07	18.74	25.35	6.68	23.39	24.99	5.97	23.24	25.74	7.32	23.99	25.14	6.02	23.39	**26.89**	**8.13**	**24.37**	41.95	14.31	38.28
Company	23.01	3.97	20.92	22.92	3.70	20.65	19.24	1.96	17.59	24.82	4.70	22.73	22.28	4.08	20.50	25.30	5.07	23.28	22.80	4.35	21.27	**27.08**	**9.44**	**25.37**	40.20	12.30	36.16
EduIns.	22.12	3.95	19.85	21.47	4.29	19.24	18.27	2.95	15.83	25.59	7.58	23.83	24.17	6.70	21.96	26.05	8.74	24.03	25.01	6.96	22.75	**28.35**	**9.11**	**26.10**	39.11	14.04	35.18
Event	26.98	5.72	24.53	26.64	5.67	24.08	22.75	3.02	20.74	28.33	7.72	26.07	28.31	7.69	26.20	28.96	8.01	26.49	28.30	7.76	26.13	**29.27**	**8.72**	**27.88**	46.17	16.90	41.87
Film	21.57	3.67	19.79	21.25	3.81	19.14	17.52	1.73	15.62	22.84	5.75	20.27	20.58	5.34	18.86	22.91	7.02	20.43	21.34	5.82	20.57	**23.67**	**7.64**	**21.54**	40.24	13.78	36.14
Group	23.47	3.52	21.72	23.30	3.62	20.20	19.79	1.51	17.60	25.82	4.79	23.53	25.51	4.97	23.51	27.05	6.02	24.96	26.06	5.33	23.74	**28.04**	**7.95**	**25.38**	41.36	13.23	37.56
HisPlace.	19.29	3.49	17.01	18.96	3.71	17.51	15.67	1.78	13.25	26.89	7.13	24.72	27.40	8.08	25.69	29.01	8.95	26.37	28.12	8.19	25.77	**30.11**	**9.63**	**28.76**	37.78	10.83	34.65
Infra.	21.55	4.82	19.32	20.40	3.27	18.39	17.96	2.13	15.44	26.24	8.37	24.41	27.86	9.24	25.80	27.94	9.66	24.97	27.53	9.31	25.82	**28.43**	**9.75**	**26.62**	36.04	10.00	32.25
MOTrans.	21.89	4.05	19.73	21.20	3.93	19.31	18.28	2.37	16.16	23.33	6.47	21.62	24.52	7.04	22.72	25.53	7.93	23.06	27.95	9.41	25.81	**26.16**	**8.03**	**23.23**	41.13	13.70	37.45
OffHolder.	19.21	3.52	17.26	18.45	3.15	16.77	15.78	1.62	13.18	20.58	5.98	18.02	19.63	5.24	18.12	22.57	6.97	21.02	21.34	6.24	19.97	**23.62**	**7.13**	**21.57**	39.60	14.70	36.04
Plant	19.84	3.87	17.46	18.73	3.02	16.84	14.81	1.54	12.05	24.23	6.49	22.76	25.29	6.30	23.20	26.02	7.03	24.12	25.17	6.45	23.41	**26.65**	**7.94**	**24.51**	34.93	9.66	31.31
Single	18.79	3.81	16.24	17.96	2.67	15.86	15.25	1.78	13.27	22.37	6.75	20.26	22.06	6.78	19.98	24.43	8.02	21.98	22.84	7.02	20.52	**26.81**	**8.97**	**23.61**	36.51	11.57	31.88
SocPlayer.	16.33	2.18	14.23	14.79	2.36	12.89	13.56	1.63	11.32	19.14	3.87	17.39	12.89	1.86	12.05	20.18	5.36	18.82	14.27	2.53	14.72	**25.25**	**6.15**	**23.41**	31.06	8.00	27.08
Software	25.85	5.12	23.19	24.54	4.56	22.05	20.96	2.89	18.51	23.92	4.91	21.02	20.51	5.15	18.82	24.22	5.90	21.94	22.71	5.48	19.62	**27.47**	**6.29**	**23.27**	42.79	13.96	38.30
TelShow.	20.42	4.03	18.54	19.77	3.21	17.684	17.21	1.90	15.71	20.58	4.52	18.05	19.20	3.53	17.42	22.19	4.92	19.92	21.43	3.96	17.65	**25.27**	**5.95**	**22.56**	40.35	13.47	35.67
Town	17.95	3.61	15.72	17.89	3.56	16.50	14.77	1.72	12.23	19.52	4.96	17.43	17.96	4.39	16.87	27.79	9.06	25.07	21.37	6.47	19.88	**29.64**	**9.19**	**27.63**	33.21	10.31	30.70
WriWork.	22.49	4.31	20.34	23.39	3.89	21.14	20.75	1.62	18.80	21.83	4.47	19.04	22.19	4.33	20.15	23.81	5.06	20.33	22.38	4.90	20.18	**25.56**	**5.85**	**23.25**	42.66	13.93	38.16
AVG	21.15	4.47	18.94	20.47	3.59	18.45	17.58	2.05	15.33	23.76	5.93	21.63	22.94	5.74	21.02	25.35	7.11	23.02	23.82	6.24	21.95	**27.23**	**8.00**	**24.97**	38.95	12.64	35.03

summarizer with fine-tuned bidirectional encoder representation from transformers (BERT) as the source encoder. We denote PreSumm with RoBERTaCLS and AspCLS as PreSumm$_1$ and PreSumm$_2$, respectively. We also provide the result of Longformer [33] with RoBERTaCLS.

D. Automatic Evaluation

1) Aspect Discovery: Following previous work, we report precision, recall, and $F1$ as the automatic evaluation metrics for aspect discovery. We extract the aspects existing in gold standard summaries as the ground truth of aspect discovery. In Table III, we report the results on the WikiAsp test set, and our proposed model AspCLS significantly outperforms the baseline models on all domains.

Our aspect discovery model achieves average scores of 63.36%, 57.46%, and 59.91% on the three metrics, which verifies the effectiveness of the proposed method. Compared with the baseline classifiers, we see a general trend of high-precision predictions made by the model. We can see that the classifier performed best with the Town domain by achieving the highest precision and the $F1$ score, 33.48% better than the baseline model on $F1$. Besides, the classifier has the most significant improvement in the Infrastructure domain by 38.88%. The AspCLS model is statistically significant (using student t-test; $p < 0.05$).

2) Aspect-Based Summarization: Following [9], we focus on evaluating the model's ability to summarize inputs, particularly on the aspects existing in the gold standard summaries. Specifically, generated summaries are paired with corresponding reference summaries with the same aspects. We report ROUGE-1 (unigram), ROUGE-2 (bigram), and ROUGE-L (longest common subsequence) scores as the metrics for automatic evaluation [45]. In addition, due to the lack of training data for some domains, we use TF-IDF to calculate similarity and add the selected sentences as training data to enhance the learning ability of the model.

In Table IV, we report the results on the WikiAsp test set, and our proposed AspSumm outperforms various previous models on all domains. Our abstractive summarization model achieves average scores of 27.23, 8.00, and 24.97 on the three ROUGE metrics. The PreSumm$_2$ model performs better than the PreSumm$_1$ model by 1.36 on ROUGE-2 $F1$. We attribute this result to the observation that better classification helps generate higher-quality summaries.

Among the abstractive baselines, PreSumm$_2$ performs much better than SF and achieves an improvement of 1.18 points on the ROUGE-2 $F1$, which demonstrates the superiority of the Transformer architecture. Our abstractive aspect-based model gains an improvement of 0.89 points compared with PreSumm$_2$, 2.07 points compared with SF, and 2.26 points compared with PreSumm$_1$ on ROUGE-2 $F1$, which verifies the effectiveness of the proposed two-stage general framework for the multidocument aspect-based summary generation. The same conclusion can be found in the comparison of the results of Longformer and AspSumm.

The upper bound of model performance is described in Table IV, which chooses sentences directly from cited reference texts to maximize the ROUGE score against summaries. The gap between the extractive oracle model and other methods indicates the importance of accurate content selection before summarization, and more efforts can be made to improve content selection.

E. Human Evaluation

To evaluate the linguistic quality of generated aspect-specific summaries, we carry out a human evaluation that is a blind test. We focus on three dimensions: **correlation**, **fluency**, and **informativeness**. The correlation indicator measures whether the summary is related to the aspect. The fluency indicator can reflect the readability of generated summaries. The informativeness indicator focuses on whether the summaries cover the salient information. We sample 200 instances from five domains of the WikiAsp test set and employ ten graduate students to rate each summary. Each sample will be judged by everyone, and the final scores are the average of all corresponding judges.

Results are presented in Fig. 4. We can see that our model performs much better than all baselines. In the correlation indicator, our model achieves a high score of 3.44, which is higher than 3.14 of SF, 2.68 of PreSumm$_1$, and 2.74 of PreSumm$_2$, indicating that our model can generate summaries more relevant to the target aspects. In the fluency indicator,

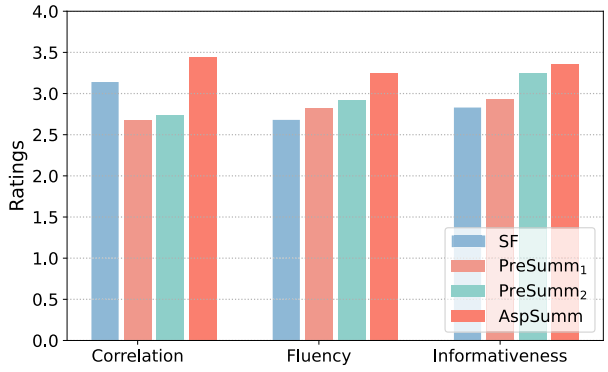


Fig. 4. Human evaluation. The summaries are rated on a Likert scale of 1 (worst) to 5 (best). All models are significantly different from AspSumm (using a paired student t-test; $p < 0.05$).

TABLE V

RESULTS OF ABLATION STUDY ON THE WIKIASP DEVELOPMENT SET. OUR PROPOSED TMT STRATEGY IMPROVES THE PERFORMANCE, ESPECIALLY WHEN PARTIAL ASPECTS ARE AVAILABLE. AND THE INDUCED ASPECT INFORMATION IMPROVES THE PERFORMANCE OF ASPSUMM. ALL MODEL VARIANTS OF ASPSUMM ARE STATISTICALLY SIGNIFICANT (USING STUDENT T-TEST; $p < 0.05$)

	AspCLS		AspSumm		
	0%	50%	R-1	R-2	R-L
Artist	45.13	46.02	26.43	6.45	24.18
w/o doc	5.46	17.37	-	-	-
w/o TMT	41.80	43.15	-	-	-
w/o asp	-	-	24.22	5.67	22.18
HistoricPlace	77.09	77.48	30.11	9.63	28.76
w/o doc	6.57	25.25	-	-	-
w/o TMT	74.61	75.32	-	-	-
w/o asp	-	-	28.72	8.41	26.14
Town	85.55	85.89	29.64	9.19	27.63
w/o doc	6.41	27.92	-	-	-
w/o TMT	83.26	84.31	-	-	-
w/o asp	-	-	26.71	8.89	24.14

our model is 0.27 better than PreSumm_2. It indicates that our model can improve the readability of the summary. Our model outperforms all baselines in the informativeness indicator, which indicates our model can effectively capture salient information. All differences between AspSumm and comparison models are statistically significant (using t-test, $p < 0.05$).

F. Ablation Study

We perform an ablation study on the development set to investigate the influence of different modules in our proposed AspSumm model. For aspect discovery, we examine two settings: regular inference (equivalent to 0% known aspects) and 50% known aspects inference. For AspCLS (without doc), we remove the text features **S** and predict unknown aspects given only known aspects. This experiment tells us how much information the model can learn just from aspects. For AspCLS (without TMT), we remove the TMT procedure to test the effectiveness of the technique. More specifically, we assume all aspects are unknown during training and set all tag embeddings **T** as *unknown*. As without doc and without TMT are mainly tested for AspCLS, so we do not consider the result for AspSumm. For AspSumm, we remove the induced aspect state embeddings to test the influence of the learned sharing aspect information.

We randomly sample three domains: the Artist domain, the HistoricPlace domain, and the Town domain. Table V presents the results. In which we report precision for AspCLS and ROUGE for AspSumm. We find that even without text features, AspCLS can effectively learn rich dependencies from aspect annotations. For the setting without the TMT strategy, the performance drops in three domains. The results indicate that the model benefits from the TMT strategy. On the one hand, it can learn latent dependencies through the comparison between the two settings. On the other hand, it can improve prediction accuracy from provided partially known aspects.

In addition, we find that without sharing embeddings, the ROUGE-2 $F1$ score of abstractive summarization drops by

TABLE VI

SAMPLE SUMMARIES FOR A DOCUMENT CLUSTER FROM THE WIKIASP TEST SET. ARTICLE FROM THE TOWN DOMAIN WITH ASPECTS GEOGRAPH, NAME, HISTORY, AND GOVERNMENT

Domain/Title: Town/Jefferson Township, Richland County, Ohio
Aspect: Geograph
Gold: located in the southern part of the county , it borders the following townships : washington township - north monroe township - northeast corner worthington township - east pike township , knox county - southeast corner berlin township , knox county - south middlebury township , knox county - southwest perry township - west troy township - northwest corner part of the village of bellville is located in northern jefferson township . **AspSumm:** located in the southern part of the county, it borders the following townships : washington township - north monroe township - northeast corner harrison township - southeast corner hancock township - south franklin township - west no municipalities are located in london township, although the unincorporated community of the center is located in the township.
Aspect: Name and history
Gold: it is one of twenty - four jefferson townships statewide . **AspSumm:** it is one of twenty - four jefferson townships statewide, although there is a part of the township in perry county, and a small portion in the area of the west part of perry township in the township of the county seat in the county.
Aspect: Government
Gold: the township is governed by a three - member board of trustees , who are elected in november of odd - numbered years to a four - year term beginning on the following january 1 . two are elected in the year after the presidential election and one is elected in the year before it . there is also an elected township fiscal officer , who serves a four - year term beginning on april 1 of the year after the election , which is held in november of the year before the presidential election . vacancies in the fiscal officership or on the board of trustees are filled by the remaining trustees. **AspSumm:** the township is governed by a three - member board of trustees, who are elected in november of odd - numbered years to a four - year term beginning on the following january 1. of the year after the presidential election and one is elected in the year before it, who serves a four, which is also an elected township fiscal officer, is held in no

1.38, 0.18, and 2.30 for the Artist domain, the HistoricPlace domain, and the Town domain, respectively. It indicates aspect representation is necessary to improve the performance of abstractive AspSumm.

G. Case Study

In Table VI, we present example summaries generated by our multidocument aspect-based method. Given multi-document input, the model correctly identifies the contained aspects in the documents and generates relevant summaries. Furthermore, the model helps generate full English Wikipedia articles, which have great practical significance. It is noticeable that our approach identifies predefined aspects and generates precise summaries related to aspects. As Wikipedia editors usually convert the text into more encyclopedic text with a unified format, our model with aspect information can learn the structure and generate high overlapping summaries with the ground truth. However, the problem of content deviation still needs to be improved.

V. Conclusion and Future Work

In this work, we propose a two-stage general framework for multidocument AspSumm. It exploits the latent dependencies by utilizing the TMT strategy and inducing the complex correlations to the abstractive summarization model. Experimental results show that the proposed method significantly outperforms the baseline methods on classification and summarization and achieves the best result on the WikiAsp dataset. Both automatic evaluation and human evaluation indicate that our proposed model improves aspect relevance, resulting in higher-quality summaries.

In the future, we plan to improve the performance of AspCLS by using hierarchical architecture and exploring the design of better training strategies to make AspCLS generalize to settings where some aspects have never been observed in training. We also plan to explore a model for joint aspect discovery and aspect-related summary generation.

References

[1] 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.

[2] Z. Hao, J. Ji, T. Xie, and B. Xue, "Abstractive summarization model with a feature-enhanced Seq2Seq structure," in *Proc. 5th Asia–Pacific Conf. Intell. Robot Syst. (ACIRS)*, Jul. 2020, pp. 163–167.

[3] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1073–1083.

[4] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1797–1807.

[5] H. Jin, T. Wang, and X. Wan, "Multi-granularity interaction network for extractive and abstractive multi-document summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6244–6254.

[6] L. Frermann and A. Klementiev, "Inducing document structure for aspect-based summarization," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 6263–6273.

[7] K. Krishna and B. V. Srinivasan, "Generating topic-oriented summaries using neural attention," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 1697–1705.

[8] B. Tan, L. Qin, E. Xing, and Z. Hu, "Summarizing text on any aspects: A knowledge-informed weakly-supervised approach," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 6301–6309.

[9] H. Hayashi, P. Budania, P. Wang, C. Ackerson, R. Neervannan, and G. Neubig, "WikiAsp: A dataset for multi-domain aspect-based summarization," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 211–225, Mar. 2021.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.

[11] I. Titov and R. McDonald, "A joint model of text and aspect ratings for sentiment summarization," in *Proc. ACL HLT*, Columbus, OH, USA, Jun. 2008, pp. 308–316. [Online]. Available: https://www.aclweb.org/anthology/P08-1036

[12] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in *Proc. 18th Int. Conf. World wide web*, Apr. 2009, pp. 131–140.

[13] L. Wang and W. Ling, "Neural network-based abstract generation for opinions and arguments," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, San Diego, CA, USA, Jun. 2016, pp. 47–57. [Online]. Available: https://www.aclweb.org/anthology/N16-1007

[14] M. Yang, Q. Qu, J. Zhu, Y. Shen, and Z. Zhao, "Cross-domain aspect/sentiment-aware abstractive review summarization," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Santa Fe, NM, USA, Oct. 2018, pp. 1110–1120, [Online]. Available: https://www.aclweb.org/anthology/C18-1095

[15] S. Angelidis and M. Lapata, "Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3675–3686.

[16] A. Bražinskas, M. Lapata, and I. Titov, "Unsupervised opinion summarization as copycat-review generation," 2019, *arXiv:1911.02247*.

[17] R. K. Amplayo and M. Lapata, "Unsupervised opinion summarization with noising and denoising," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1934–1945.

[18] R. Mukherjee, H. C. Peruri, U. Vishnu, P. Goyal, S. Bhattacharya, and N. Ganguly, "Read what you need: Controllable aspect-based opinion summarization of tourist reviews," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1825–1828.

[19] Y. Zhang, E. Chen, and W. Xiao, "Extractive-abstractive summarization with pointer and coverage mechanism," in *Proc. Int. Conf. Big Data Technol.*, 2018, pp. 69–74.

[20] M. T. R. Laskar, E. Hoque, and J. Huang, "Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models," in *Proc. Can. Conf. Artif. Intell.* Cham, Switzerland: Springer, 2020, pp. 342–348.

[21] N. Iskender, T. Polzehl, and S. Möller, "Towards a reliable and robust methodology for crowd-based subjective quality assessment of query-based extractive text summarization," in *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 245–253.

[22] H. T. Dang, "Overview of DUC 2005," in *Proc. Document Understand. Conf.*, 2005, pp. 1–12.

[23] Z. Cao, W. Li, S. Li, F. Wei, and Y. Li, "AttSum: Joint learning of focusing and summarization with neural attention," in *Proc. COLING*, 2016, pp. 547–556.

[24] E. Egonmwan, V. Castelli, and M. A. Sultan, "Cross-task knowledge transfer for query-based text summarization," in *Proc. 2nd Workshop Mach. Reading Question Answering*, 2019, pp. 72–77.

[25] P. J. Liu et al., "Generating Wikipedia by summarizing long sequences," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–18.

[26] A. Fan, C. Gardent, C. Braud, and A. Bordes, "Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 4177–4187.

[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] S. Narayan, N. Papasarantopoulos, S. B. Cohen, and M. Lapata, "Neural extractive summarization with side information," 2017, *arXiv:1704.04530*.

[29] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

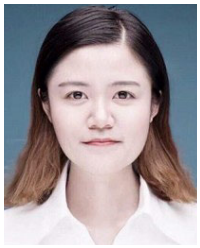
[30] E. Egonmwan and Y. Chali, "Transformer-based model for single documents neural summarization," in *Proc. 3rd Workshop Neural Gener. Transl.*, 2019, pp. 70–79.
[31] Q. Grail, J. Perez, and E. Gaussier, "Globalizing BERT-based transformer architectures for long document summarization," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021, pp. 1792–1810.
[32] Z. Wang et al., "Friendly topic assistant for transformer based abstractive summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 485–497.
[33] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020, *arXiv:2004.05150*.
[34] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[35] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3721–3731.
[36] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 115–124.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.
[38] J. Shen, X. Zhang, B. Hu, G. Wang, and Z. Ding, "An improved empirical mode decomposition of electroencephalogram signals for depression detection," *IEEE Trans. Affect. Comput.*, vol. 13, no. 1, pp. 262–271, Jan./Mar. 2022.
[39] J. Shen et al., "An optimal channel selection for EEG-based depression detection via kernel-target alignment," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 7, pp. 2545–2556, Jul. 2021.
[40] J. Shen et al., "Exploring the intrinsic features of EEG signals via empirical mode decomposition for depression recognition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 356–365, 2023.
[41] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.
[42] S. Edunov, A. Baevski, and M. Auli, "Pre-trained language model representations for language generation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 4052–4059.
[43] R. You, Z. Zhang, Z. Wang, S. Dai, H. Mamitsuka, and S. Zhu, "AttentionXML: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2020, pp. 1–11.
[44] W.-C. Chang, H.-F. Yu, K. Zhong, Y. Yang, and I. S. Dhillon, "Taming pretrained transformers for extreme multi-label text classification," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3163–3171.
[45] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out, Post Conf. Workshop ACL*, 2004, pp. 74–81.

Ye Wang received the master's degree from Jilin University, Changchun, China, in 2018. She is currently pursuing the Ph.D. degree with the National University of Defense Technology, Changsha, China.

Her current research interests include text summarization and natural language generation.

Yingmin Zhou received the master's degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2018.

She is currently a Software Development Engineer with Baidu Inc., Beijing. Her research interests include machine learning, data mining, and recommendation systems.



Mengzhu Wang received the master's degree from Chongqing University, Chongqing, China, in 2018. She is currently pursuing the Ph.D. degree with the Science and Technology on Parallel and Distributed Laboratory, School of Computer Science, National University of Defense Technology, Changsha, China, in 2022.

She is currently a Visiting Scholar with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her current research interests include transfer learning and computer vision.

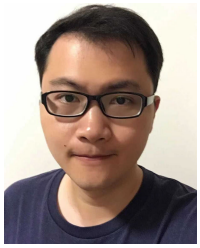
Zhenghan Chen (Graduate Student Member, IEEE) received the master's degree from Peking University, Beijing, China, in 2022.

He is currently a Kaggle Master and a Staff Member doing research on artificial intelligence recommendation algorithms with Microsoft, Beijing. His current research interests include graph representation learning and natural language processing.

Zhiping Cai received the B.Eng., M.A.Sc., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, China, in 1996, 2002, and 2005, respectively.

He is currently a Full Professor with the College of Computer, NUDT. His current research interests include artificial intelligence, network security, and big data.

He is also a Distinguished Member of the China Computer Federation (CCF).



Junyang Chen (Member, IEEE) received the Ph.D. degree in computer and information science from the University of Macau, Macau, China, in 2020.

He is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include graph neural networks, text mining, and recommender systems.

Victor C. M. Leung (Life Fellow, IEEE) is currently a Distinguished Professor of computer science and software engineering with Shenzhen University, Shenzhen, China. He is also an Emeritus Professor of electrical and computer engineering and the Director of the Laboratory for Wireless Networks and Mobile Systems with The University of British Columbia (UBC), Vancouver, BC, Canada. His research interests include wireless networks and mobile systems.