Abstractive Meeting Summarization via Hierarchical Adaptive Segmental Network Learning

Zhou Zhao*†
Zhejiang University
zhaozhou@zju.edu.cn

Haojie Pan*
The Hong Kong University of Science and Technology
hpanad@cse.ust.hk

Changjie Fan
NetEase Fuxi AI Lab
fanchangjie@corp.netease.com

Yan Liu
University of Southern California
yanliu.cs@usc.edu

Linlin Li
Alibaba Group
linyan.lll@alibaba-inc.com

Min Yang
Shenzhen Institutes of Advanced Technology
min.yang@siat.ac.cn

Deng Cai
State Key Lab of CAD & CG, Zhejiang University
dengcai@gmail.com

ABSTRACT

Abstractive meeting summarization is a challenging problem in natural language understanding, which automatically generates the condensed summary covering the important points in the meeting conversation. However, the existing abstractive summarization works mainly focus on the structured text documents, which may be ineffectively applied to the meeting summarization task due to the lack of modeling the unstructured long-form conversational contents. In this paper, we consider the problem of abstractive meeting summarization from the viewpoint of hierarchical adaptive segmental encoder-decoder network learning. We propose the hierarchical neural encoder based on adaptive recurrent networks to learn the semantic representation of meeting conversation with adaptive conversation segmentation. We then develop the reinforced decoder network to generate the high-quality summaries for abstractive meeting summarization. We conduct the extensive experiments on the well-known AMI meeting conversation dataset to validate the effectiveness of our proposed method.

CCS CONCEPTS

• **Information systems** → **Retrieval tasks and goals**; **Information retrieval**; **Summarization**.

KEYWORDS

Abstractive Meeting Summarization, Adaptive Network Learning, Neural Networks

*These two authors contributed equally to the paper.
†Corresponding author

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313619

ACM Reference Format:
Zhou Zhao, Haojie Pan, Changjie Fan, Yan Liu, Linlin Li, Min Yang, and Deng Cai. 2019. Abstractive Meeting Summarization via Hierarchical Adaptive Segmental Network Learning. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313619

1 INTRODUCTION

Abstractive meeting summarization is a challenging task in natural language understanding, which could be of great value to the users by providing quick access to the important content of past meetings [24]. The essential problem of abstractive meeting summarization is to automatically generate the condensed summary covering the important points in the meeting conversation. In order to provide the high-quality summaries, many existing works learn the semantic representation to capture the overall content of the conversation for abstractive meeting summarization. Currently, the existing meeting summarization works [2, 10, 12, 16, 21, 24] generate the meeting summaries mainly based on the extracted semantic rules, which suffer from the low maintainability [23]. On the other hand, most of the existing abstractive summarization approaches [3, 14, 26] mainly focus on the structured text documents, which learn the semantic representation of text documents and then generate the summaries for abstractive document summarization. Although these works have achieved great performance in abstractive document summarization, they may still be ineffectively applied to meeting summarization due to the lack of modeling the unstructured long-form meeting conversation contents.

The meeting conversation contents are the sequential collection of unstructured utterances, which have long-term semantic dependencies [24]. Recently, the hierarchical neural encoder [8] has been proposed to learn the paragraph-level semantic representation for document modeling. We then employ the hierarchical neural encoder to learn the high-level semantic representation that models the long-term semantic dependencies in the meeting conversation contents. On the other hand, although the sequential utterances in

the meeting conversation are topically consistent, they are unstructured and have different semantic contents. To tackle this issue, we employ the binary neurons [4] and then develop the hierarchical recurrent encoder based on adaptive binary neural networks. The adaptive hierarchical encoder network then learns the high-level semantic representation of meeting conversation with adaptive conversation segmentation. Thus, leveraging hierarchical neural encoder with adaptive binary neural networks is important for modeling the unstructured long-form meeting conversation contents for abstractive meeting summarization.

In this paper, we present the problem of abstractive meeting summarization from the viewpoint of hierarchical adaptive segmental encoder-decoder network learning. We first propose the hierarchical neural encoder with adaptive recurrent networks to learn the high-level semantic representation of meeting conversation contents with adaptive conversation segmentation. We then develop the the reinforced decoder networks to generate the high-quality summaries for abstractive meeting summarization. We next devise the Hierarchical encoder-decoder network learning framework with Adaptive conversation Segmentation for abstractive meeting summarization, named as HAS. When a certain meeting conversation is issued, HAS can automatically generate the natural language summaries for it based on its conversation contents. The main contributions of this paper are as follows:

- Unlike the previous studies, we study the problem of abstractive meeting summarization from the viewpoint of hierarchical encoder-decoder network learning framework with adaptive conversation segmentation.
- We propose the hierarchical adaptive encoder to learn the high-level semantic representation of meeting conversation contents, and then devise the reinforced decoder network to generate the summaries for abstractive meeting summarization.
- We validate the effectiveness of our proposed method through extensive experiments on the well-known AMI meeting conversation dataset.

The rest of this paper is organized as follows. In Section 2, we introduce the problem of abstractive meeting summarization from the viewpoint of hierarchical encoder-decoder network learning framework with adaptive conversation segmentation. A variety of experimental results are presented in Section 3. We provide a brief review of the related work about meeting summarization and abstractive document summarization in Section 4. Finally, we provide some concluding remarks in Section 5.

2 ABSTRACTIVE MEETING SUMMARIZATION VIA HIERARCHICAL ADAPTIVE SEGMENTAL NETWORKS

In this section, we first present the problem of abstractive meeting summarization from the viewpoint of hierarchical adaptive segmental encoder-decoder network learning framework. We then introduce the hierarchical adaptive segmental encoder networks that learn the high-level semantic representation of meeting conversation contents. We next illustrate the reinforced decoder networks to generate the summaries for abstractive meeting summarization.

2.1 The Problem

Before presenting the learning framework, we first introduce some basic notions and terminologies. We denote the utterances in meeting conversation by $\mathbf{u} \in U$ and its meeting summary by $\mathbf{z} \in Z$, where U and Z are the sets of meeting conversations and meeting summaries. The sequential collection of utterances in meeting conversation are denoted by $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n)$, where $\mathbf{u}_i$ is the i-th utterance and n is the number of utterances in meeting conversation $\mathbf{u}$. The word-level representation of utterance $\mathbf{u}_i$ is denoted by $\mathbf{u}_i = (\mathbf{u}_{(i,1)}, \mathbf{u}_{(i,2)}, \ldots, \mathbf{u}_{(i,m_i)})$, where $\mathbf{u}_{(i,j)}$ is the embedding vector for the j-th word and m_i is the number of the words in utterance $\mathbf{u}_i$. The word-level representation of meeting summary $\mathbf{z}$ is given by $\mathbf{z} = (z_1, z_2, \ldots, z_{l_z})$, where z_j is the j-th word and l_z is the number of words in summary $\mathbf{z}$. Since the meeting conversation and its utterances are sequential data with variant length, it is natural to choose the variant recurrent neural network called long-short term memory network (LSTM) [7] to learn their feature representations by

$$\mathbf{i}_t = \delta(\mathbf{W}_i \mathbf{x}_t + \mathbf{G}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \quad (1)$$

$$\hat{\mathbf{c}}_t = tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{G}_t \mathbf{h}_{t-1} + \mathbf{b}_f) \quad (2)$$

$$\mathbf{f}_t = \delta(\mathbf{W}_f \mathbf{x}_t + \mathbf{G}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \quad (3)$$

$$\mathbf{c}_t = \mathbf{i}_t \cdot \hat{\mathbf{c}}_t + \mathbf{f}_t \cdot \mathbf{c}_{t-1} \quad (4)$$

$$\mathbf{o}_t = \delta(\mathbf{W}_o \mathbf{x}_t + \mathbf{G}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \cdot tanh(\mathbf{c}_t) \quad (6)$$

where δ represents the sigmoid activation function; $\mathbf{W}$s, $\mathbf{G}$s are the weight matrices, and $\mathbf{b}$s are the bias vectors. The memory cell $\mathbf{c}_t$ maintains the history of the inputs observed up to the timestep. Update operations on the memory cell are modulated by three gates $\mathbf{i}_t$, $\mathbf{f}_t$ and $\mathbf{o}_t$, which are all computed as a combination of the current input $\mathbf{x}_t$ and of the previous hidden state $\mathbf{h}_{t-1}$, followed by a sigmoid activation. The input gate $\mathbf{i}_t$ controls how the current input should be added to the memory cell; the forget gate $\mathbf{f}_t$ is used to control what the cell will forget from the previous memory $\mathbf{c}_{t-1}$, and the output gate $\mathbf{o}_t$ controls whether the current memory cell should be passed as output.

Specifically, we learn the semantic representation of utterance $\mathbf{u}_i$ by word-level bi-directional LSTM networks, which consists of a forward LSTM and a backward LSTM. The backward LSTM has the same network structure with the forward one while its input sequence is reversed. We denote the hidden state of the forward LSTM for the t-th word in utterance $\mathbf{u}_i$ by $\mathbf{h}_{(i,t)}^{(w,f)}$, and the hidden state of this word in the backward LSTM by $\mathbf{h}_{(i,m_i-t)}^{(w,b)}$. Thus, we take the output of the last bi-directional LSTM cell, $\mathbf{h}_{(i,m_i)}^{(w)} = [\mathbf{h}_{(i,m_i)}^{(w,f)}, \mathbf{h}_{(i,1)}^{(w,b)}]$ as the semantic representation of utterance $\mathbf{u}_i$. The semantic representation of the sequential collection of the utterances in the meeting conversation is given by utterance-level LSTM networks, denoted by $\mathbf{h}^{(u)} = (\mathbf{h}_1^{(u)}, \mathbf{h}_2^{(u)}, \ldots, \mathbf{h}_n^{(u)})$, where $\mathbf{h}_i^{(u)}$ is the hidden state in the utterance-level LSTM network.

Using the notations above, the problem of abstractive meeting summarization is formulated as follows. Given the collection of utterances in the meeting conversation $\mathbf{u}$, our goal is to learn the

encoder-decoder network model $g(f(\mathbf{u}))$, where the encoder network $f(\mathbf{u})$ learns the semantic representation of the meeting conversation contents, and the decoder network $\hat{\mathbf{z}} = g(f(\mathbf{u}))$ generates natural language summary $\hat{\mathbf{z}}$ for abstractive meeting summarization. We present the details of the hierarchical adaptive segmental encoder-decoder network learning framework in Figure 1.

2.2 Adaptive Encoder Networks

In this section, we propose the hierarchical adaptive segmental encoder $f(\cdot)$ to learn the high-level semantic representation of meeting conversation contents with adaptive conversation segmentation.

The meeting conversation contents contain a sequential collection of unstructured utterances, which have long-term semantic dependencies [24]. Furthermore, although the utterance in the meeting conversation are topically consistent, they have different conversational contents. Inspired by binary neurons [4], we propose the hierarchical adaptive encoder networks that learn the semantic representation of meeting conversation contents with adaptive conversation segmentation. We learn the semantic representation of sequential utterance collection from utterance-level LSTM networks by $\mathbf{h}^{(u)}(\mathbf{h}_1^{(u)}, \mathbf{h}_2^{(u)}, \ldots, \mathbf{h}_n^{(u)})$. When a certain conversational topic change is estimated in the current utterance, the conversation segmentation can be done by resetting the LSTM parameters of the next utterance. To enable the conversation segmentation, we define an adaptive recurrent neural networks with binary gate function, which decides whether to transfer the LSTM parameters (i.e., hidden state $\mathbf{h}_t^{(u)}$ and memory cell $\mathbf{c}_t^{(u)}$) of current utterance to update the LSTM parameters of the next utterance (i.e., hidden state $\mathbf{h}_{t+1}^{(u)}$ and memory cell $\mathbf{c}_{t+1}^{(u)}$) by Equations (1), (2), (3), (4) and (5). Therefore, the adaptive recurrent neural networks enable the conversation segment with variable number of utterances in the encoder network. Formally, the t-th binary gate $\alpha_t(\cdot)$ is defined as a step function, which is computed as a non-linear combination of the semantic representation of the the $t+1$-th utterance from the word-level LSTM networks (i.e., $\mathbf{h}_{(t+1,m_{t+1})}^{(w)}$), and the semantic representation of the tth utterance from the utterance-level LSTM networks (i.e., $\mathbf{h}_t^{(u)}$), given by

$$\begin{aligned}&\alpha_t(\mathbf{h}_{(t+1,m_{t+1})}^{(w)}, \mathbf{h}_t^{(u)}) = \\ &1[\delta(\mathbf{w}_\alpha^T(\mathbf{W}_{\alpha w}\mathbf{h}_{(t+1,m_{t+1})}^{(w)} + \mathbf{W}_{\alpha u}\mathbf{h}_t^{(u)} \\ &+\mathbf{b}_\alpha)) > 0.5].\end{aligned}$$

The $1[\cdot]$ is a step function and $\delta(\cdot)$ is a sigmoid function. The $\mathbf{w}_g$ is a learnable row vector, $\mathbf{W}_{\alpha w}$, $\mathbf{W}_{\alpha u}$ and $\mathbf{b}_\alpha$ are the learnable weights and biases. For example, the inputs of binary gate $\alpha_2(\mathbf{h}_{3,3}^{(w)}, \mathbf{h}_2^u)$ are the semantic representation of utterance $\mathbf{h}_2^u$ from the utterance-level LSTM networks and the semantic representation of utterance $\mathbf{h}_{3,3}^{(w)}$ from the word-level LSTM networks in Figure 1. Given the sequential representation of utterances $(\mathbf{h}_1^{(u)}, \mathbf{h}_2^{(u)}, \ldots, \mathbf{h}_n^{(u)})$ with the values of binary gate $(\alpha_1, \alpha_2, \ldots, \alpha_{n-1})$, we develop the segment-level LSTM networks as follows. If the value of binary gate $\alpha_t(\cdot) = 1$, the hidden state of utterance-level LSTM networks at time $t+1$, $\mathbf{h}_{t+1}^{(u)}$, is passed as the input to the segment-level LSTM networks. For example, the hidden states $\mathbf{h}_3^{(u)}$, $\mathbf{h}_6^{(u)}$, ..., $\mathbf{h}_N^{(u)}$ in utterance-level LSTM networks are passed as the inputs to the segment-level LSTM networks in Figure 1. We denote semantic representation of the sequential segment collection from meeting conversation contents in the segment-level LSTM networks by $\mathbf{h}^{(s)} = (\mathbf{h}_1^{(s)}, \mathbf{h}_2^{(s)}, \ldots, \mathbf{h}_k^{(s)})$, where $\mathbf{h}_j^{(s)}$ is the hidden state in the segment-level LSTM network and k is the number of conversational segments. Therefore, hierarchical adaptive segmental encoder network is given by $f(\mathbf{u}) = (\mathbf{h}_1^{(s)}, \mathbf{h}_2^{(s)}, \ldots, \mathbf{h}_k^{(s)})$. We note that the hierarchical adaptive segmental encoder network learns the semantic representation of meeting conversation contents from conversational segments with variable number of utterances.

2.3 Reinforced Decoder Networks

In this section, we propose the reinforced decoder network $g(\cdot)$ based on segment-level LSTM networks to generate the natural language summary for abstractive meeting summarization.

Given the encoded semantic representation of meeting conversation contents $f(\mathbf{u}) = (\mathbf{h}_1^{(s)}, \mathbf{h}_2^{(s)}, \ldots, \mathbf{h}_k^{(s)})$ from the segment-level LSTM networks, the decoder predicts the next word in summary by sampling $z_t \sim p_\theta(z_t|\mathbf{z}_{1:t-1}, f(\mathbf{u})) = g_t(\mathbf{h}_t^{(g)}, \mathbf{v}_t, f(\mathbf{u}))$, where $g_t(\cdot)$ is the recurrent generator. The $\mathbf{h}_t^{(g)}$ is the decoder state and $\mathbf{v}_t$ is the context vector at the t-th decoder step. The context vector $\mathbf{v}_t$ is computed as the weighted sum of the encoded semantic representation from the segment-level LSTM networks by $\mathbf{v}_t = \sum_{i=1}^k \beta_{ti}\mathbf{h}_i^{(s)}$. The attention weight β_{ti} is given by $\beta_{ti} = \frac{\exp((\mathbf{h}_{t-1}^{(g)})^T\mathbf{h}_i^{(s)})}{\sum_{i=1}^k \exp((\mathbf{h}_{t-1}^{(g)})^T\mathbf{h}_i^{(s)})}$. One common approach for training the decoder network is based on the maximum likelihood estimation, given by

$$\mathcal{L}_{ML}(g(f(\mathbf{u}))) = \sum_{t=1}^{l} \log p_\theta(z_t|\mathbf{z}_{1:t-1}, f(\mathbf{u})). \tag{7}$$

We note that when computing the conditional probability $p_\theta(z_t|\mathbf{z}_{1:t-1}, f(\mathbf{u}))$, the previous word in the human-authored summaries rather than the generated word is chosen for training in maximum likelihood estimation. However, this makes the learnt decoder network suboptimal [1]. To tackle this problem, we employ the reinforcement learning framework for training the decoder network.

In the setting of reinforcement learning, we define the generation of next word as action and the decoding probability $p_\theta(\hat{z}_t|\hat{\mathbf{z}}_{1:t-1}, f(\mathbf{u}))$ as the policy. In the task of abstractive meeting summarization, the reward is usually received at the end of the generated summaries. We then choose the ROUGE [9] as the reward function $R_\mathbf{z}(\hat{\mathbf{z}})$, which is calculated by comparing the generated summary $\hat{\mathbf{z}}$ with the human-written summary $\mathbf{z}$. Specifically, we define the expected cumulative reward at each time step using value function by $Q(\hat{z}_t|\hat{\mathbf{z}}_{1:t-1}, f(\mathbf{u})) = E_{p_\theta(\hat{\mathbf{z}}_{t+1:T}|\hat{\mathbf{z}}_{1:t}, f(\mathbf{u}))}R_\mathbf{z}(\hat{\mathbf{z}})$. The value function $Q(z_t|\mathbf{z}_{1:t-1}, f(\mathbf{u}))$ is then estimated by aggregating the Monte-Carlo simulation at each time step, given by

$$\begin{aligned}&Q(\hat{z}_t|\hat{\mathbf{z}}_{1:t-1}, f(\mathbf{u})) \approx \\ &\begin{cases} \frac{1}{N}\sum_{n=1}^N R_\mathbf{z}([\hat{\mathbf{z}}_{1:t-1}, z_t, \hat{\mathbf{z}}_{t+1:l_z}^n]), & t < l \\ R_\mathbf{z}([\hat{\mathbf{z}}_{1:t-1}, \hat{z}_t]). & t = l_z \end{cases}\end{aligned} \tag{8}$$

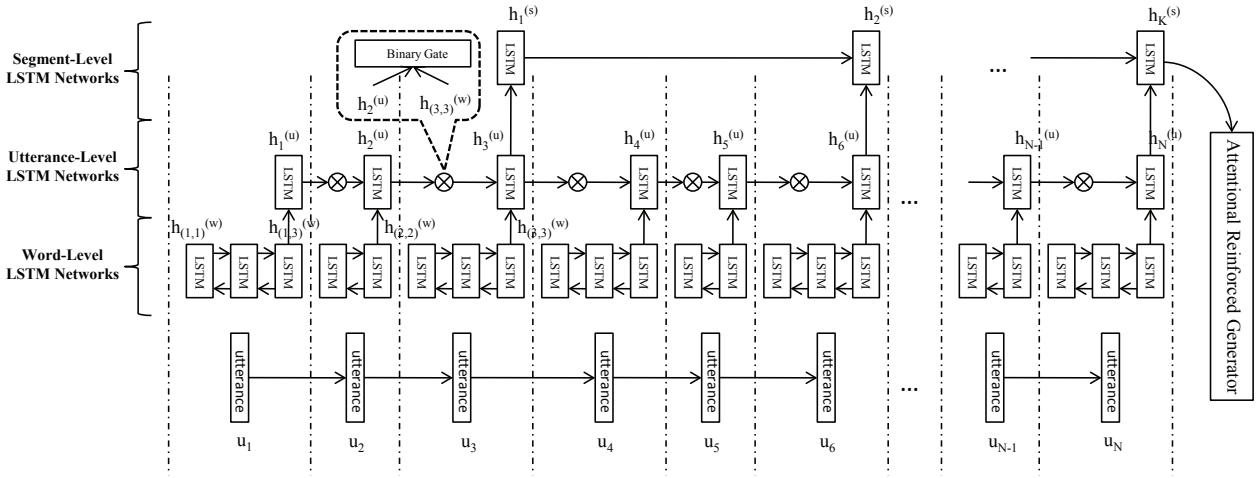

Figure 1: The Framework of Abstractive Meeting Summarization via Hierarchical Adaptive Segmental Encoder-Decoder Networks. The hierarchical encoder network learns the semantic representation of meeting conversation contents with adaptive conversation segmentation. The reinforced recurrent decoder network then generates the summaries for abstractive meeting summarization.

We denote that $\{\hat{\mathbf{z}}^1_{t+1,l_z}, \ldots, \hat{\mathbf{z}}^N_{t+1:l_z}\} \sim p_\theta(\hat{\mathbf{z}}_{t+1:l}|\hat{\mathbf{z}}_{1:t}, f(\mathbf{u}))$ as the set of simulated generated summaries, which are randomly sampled starting from the $t+1$-th time step on the current state and action. According to the policy gradient theorem, the gradients of the reinforced decoder network can be estimated by

$$\nabla_\theta \mathcal{L}_{RL}(g(f(\mathbf{u}))) = \sum_{t=1}^{l} \nabla_\theta \log p_\theta(\hat{z}_t|\hat{\mathbf{z}}_{1:t-1}, f(\mathbf{u})) Q(\hat{z}_t|\hat{\mathbf{z}}_{1:t-1}, f(\mathbf{u})). \quad (9)$$

3 EXPERIMENTS

In this section, we conduct several experiments on the AMI meeting corpus [6], to show the effectiveness of our approach for the problem of abstractive meeting summarization.

3.1 Experimental Setting

We first present the details of AMI meeting corpus dataset. The AMI meeting corpus is a collection of 142 meeting records along with their corresponding human-authored abstractive summaries. In each meeting, the group of people are engaged in a team and each speak assumes a certain role in a team. Following the experimental setting in [16, 21], we use 100 dialogs as training set, 20 dialogs as validation set and the remaining 22 dialogs as the testing set. So the training, validation and testing data do not have overlap. For evaluation, we employ ROUGE [9], which compares machine summaries with human gold-standard summaries using n-gram overlap and skip-gram overlap. We train the proposed method on machines with Linux OS, Intel(R) Core i7-5930K 3.5GHz and two GTX TITIAN X graphic cards. The initial learning rate of our method is set to 1, the decay rate is set to 0.5 and the drop rate is set to 0.5. The vocabulary size of our method is set to 10,517.

3.2 Experimental Comparisons

We evaluate the performance of our proposed method based on three widely-used evaluation for the problem of abstractive meeting summarization [10, 16], i.e., ROUGE-1, ROUGE-2 and ROUGE-SU. We compare our proposed method with other five state-of-the-art methods for the problem of abstractive meeting summarization as follows:

- **TextRank** method [11] is the extractive meeting summarization model which selects the important utterances from the meeting conversations for summarization.
- **Fusion** method [10] is the abstractive meeting summarization model which aggregates the most relevant sentences in the word graph and then selects the best paths as abstractive summaries.
- **TG** method [16] extends multi-sentence fusion algorithm to select the best templates for generating abstractive summarizes, which is based on the relationship between summaries and their source meeting transcripts.
- **AED-RNN** method [14] is under the encoder-decoder framework that generates the abstractive summaries for meeting conversation with attentional recurrent neural networks.
- **PGN** method [20] is the pointer-generator network, which is current state-of-art model in general abstractive text summarization task.
- **SEASS** method [26] is the selective encoding model based on sequence-to-sequence framework for abstractive summarization.
- **HAS-ML** method is the adaptive segmental network model that jointly segments the meeting conversation and generates the abstractive summaries, which is trained by maximum likelihood estimation.
- **HAS-RL** method is the adaptive segmental network model, which is trained by reinforcement learning.

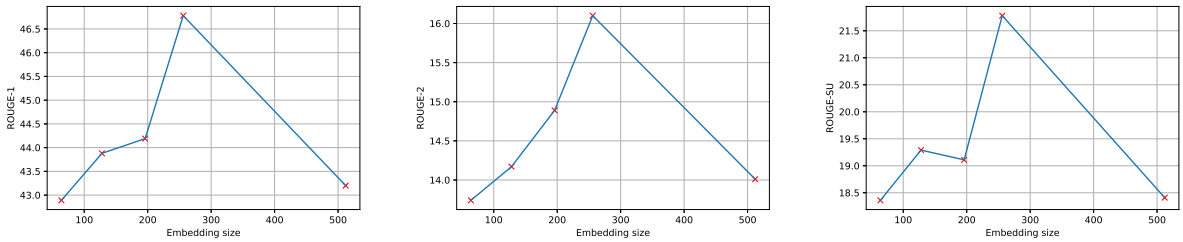

Figure 2: Effect of the dimension of word embedding on ROUGE-1, ROUGE-2 and ROUGE-SU

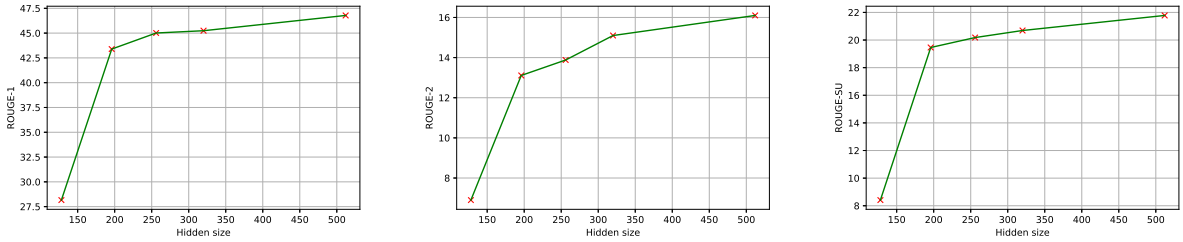

Figure 3: Effect of the dimension of LSTM hidden state on ROUGE-1, ROUGE-2 and ROUGE-SU

Table 1: Experimental results on ROUGE-1, ROUGE-2 and ROUGE-SU.

Method	ROUGE-1	ROUGE-2	ROUGE-SU
TextRank	25.12	3.37	6.61
Fusion	32.30	4.80	8.10
TG	31.50	6.70	11.40
AED-RNN	39.74	14.29	15.16
PGN	41.52	14.89	17.01
SEASS	43.93	14.29	19.10
HAS-ML	46.78	16.10	21.78
HAS-RL	**48.64**	**17.45**	**22.13**

Among them, methods TextRank, Fusion, TG, HAS-ML and HAS-RL are proposed for meeting summarization while methods AED-RNN and SEASS are designed for document summarization. The methods Fusion, TF generate the abstractive summaries for meeting conversation based on the constructed word graph. The methods ADE-RNN, PGN, and SEASS are under the encoder-decoder framework, which generates the abstractive summaries with attentional recurrent neural networks. Unlike the previous abstractive summarization methods, our HAS model jointly segments the meeting conversation and generates the abstractive summaries for meeting conversation. To exploit the effect of different training process, we denote our HAS model based on maximum likelihood by HAS-ML, and the one based on reinforcement learning by HAS-RL. We choose the ROUGE-SU evaluation criteria as the reward in the reinforcement learning for HAS-RL. The input words of our methods are initialized by pre-trained word embeddings and the weights of LSTMs are randomly by a Gaussian distribution with zero mean.

Table 1 shows the overall experimental results of the methods for abstractive meeting summarization on ROUGE-1, ROUGE-2 and ROUGE-SU, respectively. The hyperparameters and parameters are chosen to conduct the testing evaluation. We report the average value of all the methods on three evaluation criteria. The experimental results reveal a number of interesting points:

- The methods based on abstractive summarization, Fusion, TG, AED-RNN, SEASS, HAS-ML and HAS-RL outperform the extractive-based method TextRank, which suggests that the generation-based approach is critical for the problem.
- The methods based on recurrent neural networks, AED-RNN and SEASS achieve better performance than other baselines. This suggests that the deep representation of meeting conversation also improves the performance of abstractive meeting summarization.
- In all the cases, both of our HAS-ML and HAS-RL methods achieve the best performance. This fact shows that the adaptive segmental network learning framework that joint exploit the conversation segmentation and abstractive summarization generation can further improves the performance of abstractive meeting summarization.
- The HAS model trained by reinforcement learning, HAS-RL, outperforms the one trained by maximum likelihood, HAS-ML, which suggests that reinforcement learning is effective for generating the summaries in the problem of abstractive meeting summarization.

In our approach, there is two essential parameters, which are the dimension of hidden LSTM state and word embedding size. We first study the effect of word embedding on our method by varying the dimension of word embedding from 64 to 512 on ROUGE-1, ROUGE-2 and ROUGE-SU in Figures 2(a), 2(b) and 2(c). We can see that our method achieves the best performance when the size of word

Human-Authored Summary
the project manager acquainted the team with the tools and equipment around them and then had the team members introduce themselves by name and what role they had in the project. the project manager then introduced the upcoming project along with more tools and equipment to the team members. the team members then participated in an exercise in which they drew their favorite animals. after the drawing exercise, the project manager talked about the project finances and production costs. the team then discussed their experiences with remotes and various features to consider when producing a remote
Summary Generated by HAS-ML
the project manager opened the meeting and introduced the project, to design a remote control. the team members then participated in an exercise in which they each drew their favorite animal and discussed why they liked the animal. the project manager then led the team in calculating the production costs of the remote and what features they would like to include in the remote they are to create. the team discussed the interior workings of a remote and how to make the remote more technologically innovative but decided to use a spongy material in their design
Summary Generated by HAS-RL
the project manager opened the meeting and introduced the project, to design a remote control. the project manager then introduced the upcoming project to the team members and then the team members participated in an exercise in which they each drew their favorite animal and discussed what they liked about the animal. the project manager then led the team in calculating the production costs of the remote and what features they would like to include in the remote they are to create. the project manager briefed the team on some new requirements and led them in a discussion on their experiences with remotes and various features to consider in making the remote

Figure 4: A comparison between a human-authored summary and the summaries created by methods HAS-ML and HAS-RL.

embedding is set to 256. We then investigate the effect of hidden LSTM state on our method by varying the dimension of hidden LSTM state from 128 to 512 on ROUGE-1, ROUGE-2 and ROUGE-SU in Figures 3(a), 3(b) and 3(c), respectively. We can observe that the performance trend of our method becomes convergent after the dimension of hidden LSTM state greater than 256.

Finally, we demonstrate the effectiveness of the summaries generated by our HAS model with a human-authored one in Figure 4. We can observe that, compared with the human-authored summary, both the summaries generated by HAS-ML and HAS-RL methods also cover the main points in the meeting conversation well. Compared with the summary generated HAS-ML method, the summary generated by HAS-RL method are more fluent and natural.

4 RELATED WORK

In this section, we briefly review some related work on meeting summarization and abstract document summarization.

The existing work for the problem of meeting summarization can be mainly categorized as extractive-based approaches [5, 18, 25] and abstractive-based approaches [2, 10, 12, 16, 21, 24]. Xie et al. [25] treat extractive meeting summarization task as a binary classification problem. Bui et al. [5] investigate the directed graphical models for extractive meeting summarization. Riedhammer et al. [18] propose the unsupervised method for extractive meeting summarization. A recent study [13] revealed that people generally prefer abstractive summaries to the extracted ones for meeting summarization. Mehdad et al. [10] build the entailment graph for meeting summarization. Wang et al. [24] apply multiple-sequence alignment to generate templates for abstractive meeting summarization. Oya et al. [16] leverage the relationship between human-authored summaries and their source meeting transcriptions to select the templates for generating abstractive summaries for meetings. Murray et al. [12] formulate the abstractive meeting summarization as a Markov Decision Process and value iteration is used for natural language generation. Banerjee et al. [2] generate abstractive summaries by fusing important content from several utterances with dependency graph. Singla et al. [21] develop the abstractive summarization systems with automatic communities for spoken conversation summarization. Unlike previous studies, we formulate the problem of meeting summarization from the viewpoint of adaptive segmental encoding, which can be solved by recurrent neural networks with policy gradient learning.

Abstractive document summarization is the process of automatically generating natural language summaries from an input document while retaining the important points [3]. Banerjee et al. [3] tackle the abstractive summarization with integer linear programming based on word graph. Rush et al. [19] develop the neural attention-based summarization model based on neural machine translation. Tan et al. [22] propose the graph-based attention mechanism for abstractive document summarization. See et al. [20] present the hybrid pointer-generator architecture with coverage for text summarization. Paulus et al. [17] propose the neural summarization method that combines supervised word prediction and reinforcement learning. Nema et al. [15] introduce the query-based abstractive summarization task. Zhou et al. [26] develop the selective encoding for abstractive sentence summarization. However, the problem formulation of abstractive meeting summarization is different from the problem of document summarization, because of the unstructured long meeting conversation contents. Therefore, the existing abstractive document summarization methods may not be directly applied to our problem.

5 CONCLUSION

In this paper, we present the problem of abstractive meeting summarization from the viewpoint of hierarchical adaptive segmental encoder-decoder network learning framework. We first propose the adaptive segmental encoder networks that learn the semantic representation of meeting conversation contents with adaptive conversation segmentation. We then devise the reinforced decoder networks to generate the natural language summaries for abstractive meeting summarization. We evaluate the effectiveness of our method through extensive experiments on the well-known AMI meeting conversation dataset.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China under Grant No.61602405 and No.61836002, Alibaba Innovative Research, China Knowledge Centre for Engineering Sciences and Technology , Joint Research Program of ZJU and Hikvision Research Institute, and was partially funded by Microsoft Research Asia.

REFERENCES

[1] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086* (2016).

[2] Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Abstractive meeting summarization using dependency graph fusion. In *WWW*. ACM, 5–6.

[3] Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-Document Abstractive Summarization Using ILP Based Multi-Sentence Compression.. In *IJCAI*. 1208–1214.

[4] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013).

[5] Trung H Bui, Matthew Frampton, John Dowding, and Stanley Peters. 2009. Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity. In *SIGDIAL*. ACL, 235–243.

[6] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The AMI meeting corpus: A pre-announcement. In *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 28–39.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[8] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *ACL* (2015).

[9] Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. ACL, 71–78.

[10] Yashar Mehdad, Giuseppe Carenini, Frank Wm Tompa, and Raymond T Ng. 2013. Abstractive Meeting Summarization with Entailment and Fusion.. In *ENLG*. 136–146.

[11] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

[12] Gabriel Murray. 2015. Abstractive meeting summarization as a Markov decision process. In *Canadian Conference on Artificial Intelligence*. Springer, 212–219.

[13] Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2010. Generating and validating abstracts of meeting conversations: a user study. In *Proceedings of the 6th International Natural Language Generation Conference*. ACL, 105–113.

[14] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Ça glar Gulçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. *CoNLL 2016* (2016), 280.

[15] Preksha Nema, Mitesh Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven Attention Model for Query-based Abstractive Summarization. *ACL* (2017).

[16] Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *INLG*. 45–53.

[17] Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A Deep Reinforced Model for Abstractive Summarization. *arXiv preprint arXiv:1705.04304* (2017).

[18] Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tür. 2010. Long story short–global unsupervised models for keyphrase based meeting summarization. *Speech Communication* 52, 10 (2010), 801–815.

[19] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* (2015).

[20] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. *ACL* (2017).

[21] Karan Singla, Evgeny Stepanov, Ali Orkan Bayer, Giuseppe Carenini, and Giuseppe Riccardi. 2017. Automatic Community Creation for Abstractive Spoken Conversations Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. 43–47.

[22] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive Document Summarization with a Graph-Based Attentional Neural Model. In *ACL*, Vol. 1. 1171–1181.

[23] Kees van Deemter, Emiel Krahmer, and Mariët Theune. 2005. Real vs. template-based natural language generation: a false opposition. *Computational Linguistics* 31, 1 (2005), 15–24.

[24] Lu Wang and Claire Cardie. 2013. Domain-Independent Abstract Generation for Focused Meeting Summarization.. In *ACL*. 1395–1405.

[25] Shasha Xie, Yang Liu, and Hui Lin. 2008. Evaluating the effectiveness of features and sampling in extractive meeting summarization. In *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*. IEEE, 157–160.

[26] Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective Encoding for Abstractive Sentence Summarization. *ACL* (2017).