Unsupervised Abstractive Dialogue Summarization for Tete-a-Tetes

Xinyuan Zhang,[1] **Ruiyi Zhang,** [2] **Manzil Zaheer,** [3] **Amr Ahmed** [3]
[1] ASAPP
[2] Duke University
[3] Google Research
xzhang@asapp.com, ryzhang@cs.duke.edu, manzilzaheer@google.com, amra@google.com

Abstract

High-quality dialogue-summary paired data is expensive to produce and domain-sensitive, making abstractive dialogue summarization a challenging task. In this work, we propose the first unsupervised abstractive dialogue summarization model for tete-a-tetes (SuTaT). Unlike standard text summarization, a dialogue summarization method should consider the multi-speaker scenario where the speakers have different roles, goals, and language styles. In a tete-a-tete, such as a customer-agent conversation, SuTaT aims to summarize for each speaker by modeling the customer utterances and the agent utterances separately while retaining their correlations. SuTaT consists of a conditional generative module and two unsupervised summarization modules. The conditional generative module contains two encoders and two decoders in a variational autoencoder framework where the dependencies between two latent spaces are captured. With the same encoders and decoders, two unsupervised summarization modules equipped with sentence-level self-attention mechanisms generate summaries without using any annotations. Experimental results show that SuTaT is superior on unsupervised dialogue summarization for both automatic and human evaluations, and is capable of dialogue classification and single-turn conversation generation.

Introduction

Tete-a-tetes, conversations between two participants, have been widely studied as an importance component of dialogue analysis. For instance, tete-a-tetes between customers and agents contain information for contact centers to understand the problems of customers and improve the solutions by agents. However, it is time-consuming for others to track the progress by going through long and sometimes uninformative utterances. Automatically summarizing a tete-a-tete into a shorter version while retaining its main points can save a vast amount of human resources and has a number of potential real-world applications.

Summarization models can be categorized into two classes: extractive and abstractive. Extractive methods select sentences or phrases from the input text, while abstractive methods attempt to generate novel expressions which requires an advanced ability to paraphrase and condense information. Despite being easier, extractive summarization is often not preferred in dialogues for its limited capability to capture highly dependent conversation histories and produce coherent discourses. Therefore, abstractively summarizing dialogues has attracted recent research interest (Goo and Chen 2018; Pan et al. 2018; Yuan and Yu 2019; Liu et al. 2019).

Customer:	I am looking for the Hamilton Lodge in Cambridge.
Agent:	Sure, it is at 156 Chesterton Road, postcode cb41da.
Customer:	Please book it for 2 people, 5 nights beginning on Tuesday.
Agent:	Done. Your reference number is qnvdz4rt.
Customer:	Thank you, I will be there on Tuesday!
Agent:	Is there anything more I can assist you with today?
Customer:	Thank you! That's everything I needed.
Agent:	You are welcome. Any time.
Customer Summary:	i would like to book a hotel in cambridge on tuesday .
Agent Summary:	i have booked you a hotel . the reference number is qnvdz4rt . can i help you with anything else ?

Table 1: An example of SuTaT generated summaries.

However, existing abstractive dialogue summarization approaches fail to address two main problems. First, a dialogue is carried out between multiple speakers and each of them has different roles, goals, and language styles. Taking the example of a contact center, customers aim to propose problems while agents aim to provide solutions, which leads them to have different semantic contents and choices of vocabularies. Most existing methods process dialogue utterances as in text summarization without accommodating the multi-speaker scenario. Second, high-quality annotated data is not readily available in the dialogue summarization domain and can be very expensive to produce. Topic descriptions or instructions are commonly used as gold references which are too general and lack any information about the speakers. Moreover, some methods use auxiliary information such as dialogue acts (Goo and Chen 2018), semantic scaffolds (Yuan and Yu 2019) and key point sequences (Liu et al. 2019) to help with summarization, adding more burden on data annotation. To our knowledge, no previous work has focused on unsupervised deep learning for abstractive dialogue summarization.

We propose SuTaT, an unsupervised abstractive dialogue summarization approach specifically for tete-a-tetes. In this paper, we use the example of *agent* and *customer* to represent the two speakers in tete-a-tetes for better understanding. In addition to summarization, SuTaT can also be used for dialogue classification and single-turn conversation generation.

To accommodate the two-speaker scenario, SuTaT processes the utterances of a customer and an agent separately in a conditional generative module. Inspired by Zhang et al. (2019) where two latent spaces are contained in one variational autoencoder (VAE) framework, the conditional generative module includes two encoders to map a customer utterance and the corresponding agent utterance into two latent representations, and two decoders to reconstruct the utterances jointly. Separate encoders and decoders enables SuTaT to model the differences of language styles and vocabularies between customer utterances and agent utterances. The dependencies between two latent spaces are captured by making the agent latent variable conditioned on the customer latent variable. Compared to using two standard autoencoders that learn deterministic representations for input utterances, using the VAE-based conditional generative module to learn variational distributions gives the model more expressive capacity and more flexibility to find the correlation between two latent spaces.

The same encoders and decoders from the conditional generative module are used in two unsupervised summarization modules to generate customer summaries and agent summaries. Divergent from MeanSum (Chu and Liu 2019) where the combined multi-document representation is simply computed by averaging the encoded input texts, SuTaT employs a setence-level self-attention mechanism (Vaswani et al. 2017) to highlight more significant utterances and neglect uninformative ones. We also incorporate copying factual details from the source text that has proven useful in supervised summarization (See, Liu, and Manning 2017). Dialogue summaries are usually written in the third-person point of view, but SuTaT simplifies this problem by making the summaries consistent with the utterances in pronouns. Table 1 shows an example of SuTaT generated summaries.

Experiments are conducted on two dialogue datasets: MultiWOZ (Budzianowski et al. 2018) and Taskmaster (Byrne et al. 2019). It is assumed that we can only access utterances in the datasets without any annotations including dialogue acts, descriptions, instructions, etc. Both automatic and human evaluations show SuTaT outperforms other unsupervised baseline methods on dialogue summarization. We further show the capability of SuTaT on dialogue classification with generated summaries and single-turn conversation generation.

Methodology

SuTaT consists of a conditional generative module and two unsupervised summarization modules. Let $\mathbf{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ denote a set of customer utterances and $\mathbf{Y} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\}$ denote a set of agent utterances in the same dialogue. Our aim is to generate a customer summary and an agent summary for the utterances in $\mathbf{X}$ and $\mathbf{Y}$.

Figure 1 shows the entire architecture of SuTaT. Given a customer utterance $\boldsymbol{x}$ and its consecutive agent utterance $\boldsymbol{y}$, the conditional generative module embeds them with two encoders and obtain latent variables $\boldsymbol{z}_x$ and $\boldsymbol{z}_y$ from the variational latent spaces, then reconstruct the utterances from $\boldsymbol{z}_x$ and $\boldsymbol{z}_y$ with two decoders. In the latent space, the agent latent variable is conditioned on the customer latent variable; during decoding, the generated customer utterances are conditioned on the generated agent utterances. This design resembles how a tete-a-tete carries out: the agent's responses and the customer's requests are dependent on each other. The encoded utterances of a dialogue are the inputs of the unsupervised summarization modules. We employ a sentence-level self-attention mechanism on the utterances embeddings to highlight the more informative ones and combine the weighted embeddings. A summary representation is drawn from the low-variance latent space using the combined utterance embedding, which is then decoded into a summary with the same decoder and a partial copy mechanism. The whole process does not require any annotations from the data.

Conditional Generative Module

We build the conditional generative module in a SIVAE-based framework (Zhang et al. 2019) to capture the dependencies between two latent spaces. The goal of the module is to train two encoders and two decoders for customer utterances $\boldsymbol{x}$ and agent utterances $\boldsymbol{y}$ by maximizing the evidence lower bound

$$\begin{aligned}\mathcal{L}_{gen} &= \mathbb{E}_{q(\boldsymbol{z}_x|\boldsymbol{x})} \log p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}_x) - \\ &\mathrm{KL}[q(\boldsymbol{z}_x|\boldsymbol{x})||p(\boldsymbol{z}_x)] + \mathbb{E}_{q(\boldsymbol{z}_y|\boldsymbol{y},\boldsymbol{z}_x)} \log p(\boldsymbol{y}|\boldsymbol{z}_y) \\ &- \mathrm{KL}[q(\boldsymbol{z}_y|\boldsymbol{y}, \boldsymbol{z}_x)||p(\boldsymbol{z}_y|\boldsymbol{z}_x)] \leq \log p(\boldsymbol{x}, \boldsymbol{y}),\end{aligned} \tag{1}$$

where $q(\cdot)$ is the variational posterior distribution that approximates the true posterior distribution. The lower bound includes two reconstruction losses and two Kullback-Leibler (KL) divergences between the priors and the variational posteriors. By assuming priors and posteriors to be Gaussian, we can apply the reparameterization trick (Kingma and Welling 2014) to compute the KL divergences in closed forms. $q(\boldsymbol{z}_x|\boldsymbol{x})$, $q(\boldsymbol{z}_y|\boldsymbol{y}, \boldsymbol{z}_x)$, $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}_x)$, and $p(\boldsymbol{y}|\boldsymbol{z}_y)$ represent customer encoder, agent encoder, customer decoder, and agent decoder.

The correlation between two latent spaces are captured by making the agent latent variable $\boldsymbol{z}_y$ conditioned on the customer latent variable $\boldsymbol{z}_x$. We define the customer prior $p(\boldsymbol{z}_x)$ to be a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The agent prior $p(\boldsymbol{z}_y|\boldsymbol{z}_x)$ is also a Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where the mean and the variance are functions of $\boldsymbol{z}_x$,

$$\boldsymbol{\mu} = \mathrm{MLP}_{\mu}(\boldsymbol{z}_x), \quad \boldsymbol{\Sigma} = \mathrm{MLP}_{\Sigma}(\boldsymbol{z}_x).$$

This process resembles how a tete-a-tete at contact centers carries out: the response of an agent is conditioned on what the customer says.

Encoding Given a customer utterance sequence $\boldsymbol{x} = \{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_t\}$, we first encode it into an utterance embedding $\boldsymbol{e}_x$ using bidirectional LSTM (Graves, Jaitly, and Mohamed 2013) or a Transformer encoder (Vaswani et al. 2017).

The Bi-LSTM takes the hidden states $\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i; \overleftarrow{\boldsymbol{h}}_i]$ as contextual representations by processing a sequence from

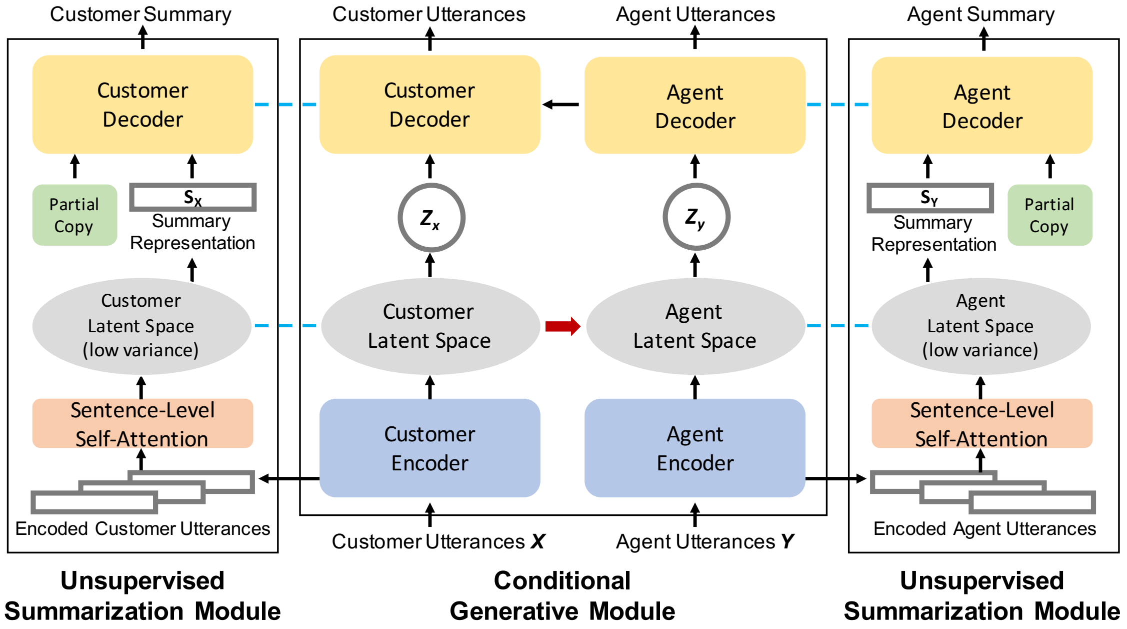

Figure 1: Block diagram of SuTaT. Architectures connected by a blue dashed line are the same. The red arrow represents the conditional relationship between two latent spaces.

both directions,

$$\overrightarrow{\boldsymbol{h}}_i = \text{LSTM}(\boldsymbol{w}_i, \boldsymbol{h}_{i-1}), \overleftarrow{\boldsymbol{h}}_i = \text{LSTM}(\boldsymbol{w}_i, \boldsymbol{h}_{i+1}).$$

The Transformer encoder produces the contextual representations that have the same dimensions as word embeddings,

$$\{\dot{\boldsymbol{w}}_1, \cdots, \dot{\boldsymbol{w}}_t\} = \text{TransEnc}(\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_t\}).$$

The customer utterance embedding $\boldsymbol{e}_x$ is obtained by averaging over the contextual representations. Similarly, we can obtain the agent utterance embedding $\boldsymbol{e}_y$.

The customer latent variable $\boldsymbol{z}_x$ is first sampled from $q(\boldsymbol{z}_x|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ using $\boldsymbol{e}_x$, then the agent latent variable $\boldsymbol{z}_y$ is sampled from $q(\boldsymbol{z}_y|\boldsymbol{y}, \boldsymbol{z}_x) = \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ using $\boldsymbol{e}_y$ and $\boldsymbol{z}_x$. The Gaussian parameters $\boldsymbol{\mu}_x$, $\boldsymbol{\Sigma}_x$, $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$ are computed with separate linear projections,

$$\boldsymbol{\mu}_x = \text{Linear}_{\mu_x}(\boldsymbol{e}_x), \boldsymbol{\mu}_y = \text{Linear}_{\mu_y}(\boldsymbol{e}_y \oplus \boldsymbol{z}_x)$$
$$\boldsymbol{\Sigma}_x = \text{Linear}_{\Sigma_x}(\boldsymbol{e}_x), \boldsymbol{\Sigma}_y = \text{Linear}_{\Sigma_y}(\boldsymbol{e}_y \oplus \boldsymbol{z}_x).$$

Decoding We first decode $\boldsymbol{z}_y$ into the agent utterance from the $p(\boldsymbol{y}|\boldsymbol{z}_y)$ using LSTM (Sutskever, Vinyals, and Le 2014) or a Transformer decoder (Vaswani et al. 2017). The decoded sequence and the latent variable $\boldsymbol{z}_x$ are then used in $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}_x)$ to generate the customer utterance.

In the LSTM decoder,

$$\boldsymbol{v}_y^{(i)} = \text{LSTM}(\boldsymbol{y}_{i-1}, \boldsymbol{z}_y, \boldsymbol{v}_y^{(i-1)})$$
$$\boldsymbol{v}_x^{(i)} = \text{LSTM}(\boldsymbol{x}_{i-1}, \boldsymbol{z}_x \oplus \boldsymbol{y}, \boldsymbol{v}_x^{(i-1)}).$$

While in the Transformer decoder,

$$\boldsymbol{v}_y^{(i)} = \text{TranDec}(\boldsymbol{y}_{<i}, \boldsymbol{z}_y)$$
$$\boldsymbol{v}_x^{(i)} = \text{TranDec}(\boldsymbol{x}_{<i}, \boldsymbol{z}_x \oplus \boldsymbol{y})$$

where $\boldsymbol{y}_{<i}$ and $\boldsymbol{x}_{<i}$ are the embeddings of the previously decoded sequence. The decoded representations $\boldsymbol{v}_y^{(i)}$ and $\boldsymbol{v}_x^{(i)}$ are put in feedforward layers to compute the vocabulary distributions,

$$p(\boldsymbol{y}_i|\boldsymbol{y}_{<i}, \boldsymbol{z}_y) = \text{softmax}(\boldsymbol{v}_y^{(i)}\mathbf{W}_y^T + \boldsymbol{b}_y)$$
$$p(\boldsymbol{x}_i|\boldsymbol{x}_{<i}, \boldsymbol{z}_x, \boldsymbol{y}) = \text{softmax}(\boldsymbol{v}_x^{(i)}\mathbf{W}_x^T + \boldsymbol{b}_x) \tag{2}$$

where $\mathbf{W}_x \in \mathbb{R}^{|x|\times l}$, $\mathbf{W}_y \in \mathbb{R}^{|y|\times l}$, $\boldsymbol{b}_x \in \mathbb{R}^l$ and $\boldsymbol{b}_y \in \mathbb{R}^l$ are learnable parameters. $|x|$ and $|y|$ are the vocabulary sizes for customer utterances and agent utterances.

Unsupervised Summarization Module

Given the encoded utterances of a dialogue, an unsupervised summarization module learns to generate a summary that is semantically similar to the input utterances using trained components from the conditional generative module.

Sentence-Level Self-Attention Some utterances like greetings or small talk do not contribute to the content of a dialogue. Therefore, we employ a sentence-level self-attention mechanism, which is built upon Multi-head attention (Vaswani et al. 2017), to highlight the most significant utterances in a dialogue.

The multi-head attention partitions the queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ into h heads along their dimensions d, and calculates h scaled dot-product attention for the linear projections of the heads.

$$\text{MH}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(head_1, \cdots, head_h)\mathbf{W}^O$$
$$head_i = \text{SDP}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

where $\mathbf{W}^O$, $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ are trainable parameters. The scaled dot-product attention outputs a weighted sum of values,

$$\mathrm{SDP}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}.$$

In SuTaT, the sentence-level self-attention is achieved by making the queries, keys, and values all be the set of encoded agent/customer utterances of a dialogue. The self-attention module assigns weights on the input utterances such that more significant and informative ones have higher weights. The output is a weighted combined utterance embedding $\tilde{\boldsymbol{e}}_X$ or $\tilde{\boldsymbol{e}}_Y$ that highlights more informative utterances from the dialogue.

Summary Generation Summary representations $\boldsymbol{s}_X$ and $\boldsymbol{s}_Y$ are sampled from the latent spaces taking the weighted combined utterance representations $\tilde{\boldsymbol{e}}_X$ and $\tilde{\boldsymbol{e}}_Y$ as inputs. To limit the amount of novelty in the generated summary, we set the variances of the latent spaces close to zero so that $\boldsymbol{s}_X \approx \boldsymbol{\mu}_x$ and $\boldsymbol{s}_Y \approx \boldsymbol{\mu}_y$. $\boldsymbol{s}_X$ and $\boldsymbol{s}_Y$ containing key information from the dialogue are decoded into a customer summary and an agent summary using the same decoders from the conditional generative module, which makes the generated summaries similar to the utterances in pronouns and language styles.

We re-encode the generated summaries into $\boldsymbol{e}_X$ and $\boldsymbol{e}_Y$ with the same encoders and compare them with each of the utterance embeddings using average cosine distance. To constrain the summaries to be semantically close to input utterances, the summarization modules are trained by maximizing a similarity loss,

$$\mathcal{L}_{sum} = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{d}(\boldsymbol{e}_X, \boldsymbol{e}_x^{(i)}) + \mathrm{d}(\boldsymbol{e}_Y, \boldsymbol{e}_y^{(i)})), \quad (3)$$

where d denotes the cosine distance.

However, the summarization modules are prone to produce inaccurate factual details. We design a simple but effective partial copy mechanism that employs some extractive summarization tricks to address this problem. We automatically make a list of factual information from the data such as dates, locations, names, and numbers. Whenever the decoder predicts a word from the factual information list, the copy mechanism replaces it with a word containing factual information from the input utterances. If there are multiple factual information words in the dialogue, the one with the highest predictive possibility will be chosen. Note that this partial copy mechanism does not need to be trained and is not activated during training.

Training Process

The objective function we optimize is the weighted sum of the reconstruction loss in Equation 1 and the similarity loss in Equation 3,

$$\mathcal{L} = \alpha\mathcal{L}_{gen} + (1-\alpha)\mathcal{L}_{sum}, \quad (4)$$

where α controls the weights of two objectives.

SuTaT involves re-encoding the generated agent utterance to help with generating the customer utterance in Equation 2 and re-encoding the generated summary to compare with utterance embeddings in Equation 3. Directly sampling from the multinomial distribution with argmax is a non-differentiable operation, so we use the soft-argmax trick (Chen et al. 2019) to approximate the deterministic sampling scheme,

$$\boldsymbol{y}_i = \mathrm{softmax}(\boldsymbol{v}_Y^{(i)}/\tau), \quad (5)$$

where $\tau \in (0, 1)$ is the annealing parameter.

Adam (Kingma and Ba 2015) is adopted for stochastic optimization to jointly train all model parameters by maximizing Equation 4. In each step, Adam samples a mini-batch of dialogues and then updates the parameters (Zhang et al. 2018).

Related Works

Dialogue Summarization Early dialogue summarization works mainly focus on extractively summarizing using statistical machine learning methods (Galley 2006; Xie, Liu, and Lin 2008; Wang and Cardie 2013). Abstractive dialogue summarization has been recently explored due to the success of sequence-to-sequence neural networks. Pan et al. (2018) propose an enhanced interaction dialogue encoder and a transformer-pointer decoder to summarize dialogues. Li et al. (2019) summarize multi-modal meetings on another encoder-decoder structure. Some approaches design additional mechanisms in a neural summarization model to leverage auxiliary information such as dialogue acts (Goo and Chen 2018), key point sequences (Liu et al. 2019), and semantic scaffolds (Yuan and Yu 2019). However, these supervised methods can only use concise topic descriptions or instructions as gold references while high-quality annotated dialogue summaries are not readily available.

Unsupervised Summarization Many extractive summarization models do not require document-summary paired data. TextRank (Mihalcea and Tarau 2004) and LexRank (Erkan and Radev 2004) encode sentences as nodes in a graph to select the most representative ones as a summary. Zheng and Lapata (2019) and Rossiello, Basile, and Semeraro (2017) advance upon TextRank and LexRank by using BERT (Devlin et al. 2019) to compute sentence similarity and replacing TF-IDF weights with word2vec embeddings respectively. In abstractive summarization, some approaches focus on learning unsupervised sentence compression with small-scale texts (Fevry and Phang 2018; Baziotis et al. 2019; West et al. 2019), while TED (Yang et al. 2020) proposes a transformer-based architecture with pretraining on large-scale data. MeanSum (Chu and Liu 2019) generates a multi-document summary by decoding the average encoding of the input texts, where the autoencoder and the summarization module are interactive. Amplayo and Lapata (2020) extends MeanSum by denoising a noised synthetic dataset. Some approaches investigate using VAE in summarization (Li et al. 2017; Schumann 2018; Bražinskas, Lapata, and Titov 2020). However, none of these methods accommodate the multi-speaker scenario in dialogues.

Model	MultiWOZ						Taskmaster					
	Customer			Agent			Customer			Agent		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
LexRank	23.54	2.63	13.43	24.35	2.79	13.29	21.64	1.83	12.86	21.54	1.90	12.15
Word2Vec	23.80	2.96	13.37	24.15	2.72	13.92	21.43	2.03	12.32	21.57	2.07	12.46
MeanSum	25.93	4.42	14.52	26.49	4.49	15.43	24.01	3.31	13.55	24.08	3.24	14.31
Copycat	26.86	4.81	16.35	26.92	4.37	16.12	24.86	4.23	14.81	25.05	3.71	15.19
VAE	26.08	4.25	14.84	26.80	3.76	15.27	24.29	3.15	14.40	24.99	3.29	14.35
SuTaT-LSTM	**28.51**	**5.60**	**17.20**	**28.71**	**5.67**	**17.49**	**26.61**	**4.89**	**16.09**	**26.67**	**4.80**	**15.74**
SuTaT-Tran	26.82	4.80	16.08	27.11	4.88	15.52	25.20	3.98	15.33	25.19	4.12	14.81
Ablation Study (with LSTM Encoders and Decoders)												
SuTaT w/o LS	24.78	3.55	14.08	25.11	4.09	14.16	23.05	3.05	13.00	23.41	3.15	13.12
SuTaT w/o Att	26.69	5.00	15.59	27.00	5.26	15.97	25.08	4.26	14.65	25.25	4.28	14.93
SuTaT w/o copy	27.65	5.23	16.01	27.67	5.47	16.42	25.28	4.80	14.97	25.15	4.47	15.16

Table 2: ROUGE scores on the MultiWOZ and Taskmaster test sets.

Experimental Details

We perform experiments with two variants of SuTaT: one equipped with LSTM encoders and decoders (SuTaT-LSTM), and the other equipped with Transformer encoders and decoders (SuTaT-Tran).

Dataset

The experiments are conducted on two dialogue datasets: MultiWOZ-2.0 (Budzianowski et al. 2018) and Taskmaster-1 (Byrne et al. 2019). MultiWOZ consists of 10438 goal-oriented human-human written dialogues between customers and agents, spanning over 7 domains such as booking hotels, booking taxis, etc. 3406 of them are single-label and 7302 of them are multi-label. The average utterances is 13.70 and average tokens per utterance is 13.82. In the experiment, we split the dataset into 8438, 1000, and 1000 dialogues for training, testing, and validation. Taskmaster consists of 7708 written dialogues created by human workers based on scenarios outlined for one of the six tasks, such as ordering pizza, ordering movie tickets, etc. The average utterances is 21.99 and average tokens per utterance is 8.62. The dataset is split into 6168, 770, and 770 dialogues for training, testing, and validation.

Baselines

To validate the effectiveness of SuTaT, we compare the two variants against the following baselines: unsupervised extractive summarization methods LexRank (Erkan and Radev 2004) and Word2Vec (Rossiello, Basile, and Semeraro 2017); unsupervised abstractive summarization methods MeanSum (Chu and Liu 2019) and Copycat (Bražinskas, Lapata, and Titov 2020). In addition, we train a vanilla text VAE model (Bowman et al. 2016) with our unsupervised summarization module as another baseline.

Since we are the first work that summarizes for each speaker in a dialogue, some modifications need to be made on baselines to make fair comparisons with our model. To make the unsupervised summarization baseline models adapt to the two-speaker scenario in tete-a-tetes, we train two models for each baseline with either customer utterances or agent utterances. During testing, the customer summaries and agent summaries are generated by the two trained models of each baseline, which are used either separately for automatic and human evaluation or concatenated together for the classification experiment.

Settings

We fine-tune the parameters of SuTaT on the validation set. VAE-based text generative models can suffer from posterior collapse where the model learns to ignore the latent variable (Bowman et al. 2016). We employ KL-term annealing and dropping out words during decoding to avoid posterior collapse. For KL annealing, the initial weights of the KL terms are 0, and then we gradually increase the weights as training progresses, until they reach the KL threshold of 0.8; the rate of this increase is set to 0.5 with respect to the total number of batches. The word dropout rate during decoding is 0.4. The latent variable size is 300 for both customer and agent latent variables. α that controls weights of two objective functions in Equation 4 is set to 0.4. The word embedding size is 300. For the bidirectional LSTM encoder and LSTM decoder, the number of hidden layers is 1 and the hidden unit size is 600. For the Transformer encoder and decoder, the number of hidden layers is 1 and the number of heads in the multi-head attention is set to 10. The number of heads in the sentence-level self-attention is also 10. The hidden unit size of the MLPs in $p(\boldsymbol{z}_y|\boldsymbol{z}_x)$ is 600. The annealing parameter τ for soft-argmax in Equation 5 is set to 0.01. During training, the learning rate is 0.0005, the batch size is 16, and the maximum number of epoch is 10. SuTaT is implemented in pytorch and trained using a NVIDIA Tesla V100 GPU with 16GB.

Reference Summaries

In this work, we define the dialogue summary as summarizing for each speaker in a dialogue and there is no such annotated dataset available. To perform summarization comparisons (Table 2 and 4), we follow the setting in (Chu and Liu 2019) to collect 200 abstractive summaries for a subset of each dataset

Model	MultiWOZ				Taskmaster			
	Customer		Agent		Customer		Agent	
	PPL	KL	PPL	KL	PPL	KL	PPL	KL
MeanSum	3.58	-	3.65	-	5.57	-	5.48	-
Copycat	3.46	0.75	3.42	0.73	5.41	0.96	5.23	0.93
VAE	3.64	0.50	3.59	0.48	5.63	0.63	5.75	0.66
SuTaT-LSTM	3.27	0.79	3.39	0.82	5.31	1.02	4.56	0.88
SuTaT-Tran	**1.77**	0.28	**2.10**	0.34	**2.48**	0.35	**2.52**	0.36

Table 3: Language modeling results on MultiWOZ and Taskmaster. Lower is better for PPL.

Model	MultiWOZ			Taskmaster		
	Info	Read	Corr	Info	Read	Corr
Reference	5.43	4.73	4.52	5.39	4.57	4.60
MeanSum	2.57	3.15	2.64	2.98	3.29	3.05
Copycat	2.89	3.37	3.00	3.04	3.49	3.07
VAE	2.96	3.04	2.44	2.97	2.92	2.45
SuTaT-LSTM	**3.68**	3.48	**4.25**	**3.61**	**3.53**	**4.20**
SuTaT-Tran	3.47	**3.56**	4.15	3.33	3.52	3.96

Table 4: Human evaluation results on informativeness, readability, and correlation of generated summaries.

using Mechanical Turk. Workers were presented with 10 dialogues from MultiWOZ and 10 dialogues from Taskmaster and asked to write summaries that “best summarize both the content and the sentiment for each speaker”. We asked workers to “write your summaries as if your were the speaker (e.g. ‘I want to book a hotel.’ instead of ‘The customer wants to book a hotel.’) and keep the length of the summary no more than one sentence”. The collected summaries are only used as reference summaries for testing and not used for model-tuning. These reference summaries cover all domains in both datasets and will be released later.

Results

We conduct the majority of experiments to show the superiority of SuTaT on unsupervised dialogue summarization. We use the labeled reference summaries for ROUGE-score-based automatic evaluation and human evaluation to compare with baseline methods. We further demonstrate the effectiveness of SuTaT by analyzing the language modeling results and using generated summaries to perform dialogue classification. In addition, we show that SuTaT is capable of single-turn conversation generation.

Unsupervised Dialogue Summarization

Automatic Evaluation ROUGE (Lin 2004) is a standard summarization metric to measure the surface word alignment between a generated summary and the reference summary. In the experiments, we use ROUGE-1, ROUGE-2, and ROUGE-L to measure the word-overlap, bigram-overlap, and longest common sequence respectively. Table 2 shows the ROUGE scores for two SuTaT variants and the baselines. As we can see, our proposed SuTaT with LSTM encoders and decoders outperforms all other baselines on both datasets. SuTaT-LSTM performs better than SuTaT-Transformer on ROUGE scores, the reason could be that Transformer decoders are too strong so the encoders are weakened during training. In general, the unsupervised abstractive models perform better than unsupervised extractive models. Compared with other unsupervised abstractive summarization baselines equipped with LSTM encoders and decoders, SuTaT-LSTM has a big performance improvement. We believe this is because SuTaT accommodates the two-speaker scenario in tete-a-tetes so that the utterances from each speaker and their correlations are better modeled.

In addition, we evaluate reconstruction performances of the language modeling based methods with perplexity (PPL), and check the posterior collapse for the VAE-based methods with KL divergence. The results for MultiWOZ and Taskmaster are shown in Table 3. As can be seen, SuTaT-Tran has much better PPL scores than other competing methods on both datasets, showing the transformer decoders are effective at reconstructing sentences. Consequently, due to the powerful decoders, SuTaT-Tran has smaller KL divergences which can lead to posterior collapse where the encoders tend to be ignored.

Human Evaluation Human evaluation for the generated summaries is conducted to quantify the qualitative results of each model. We sample 50 dialogues that are labeled with reference summaries from the MultiWOZ and taskmaster test set (25 each). With the sampled dialogues, summaries are generated from the unsupervised abstractive approaches: MeanSum, Copycat, VAE, SuTaT-LSTM, and SuTaT-Tran. We recruit three workers to rank the generated summaries and reference summaries from 6 (the best) to 1 (the worst) based on three criteria: Informativeness: a summary should present the main points of the dialogue in a concise version; Readability: a summary should be grammatically correct and well structured; Correlation: the customer summary should be correlated to the agent summary in the same dialogue.

The average ranking scores are shown in Table 4. As we can see, SuTaT-LSTM achieves the best informativeness and correlation results on both datasets while SuTaT-Tran also has good performances, further demonstrating the ability of SuTaT on generating informative and coherent dialogue summaries. In general, the two SuTaT models have better human evaluation scores than baseline models, especially on correlation scores where the results are close to reference summaries.

Model	MultiWOZ	Taskmaster
MeanSum	0.76	0.70
Copycat	0.77	0.72
VAE	0.66	0.62
SuTaT (unsupervised)	0.85	0.79
SuTaT (supervised)	0.99	0.96

Table 5: AUC scores for domain classfication with generated summaries, where MultiWOZ is multi-label and Taskmaster is single-label.

This is because SuTaT exploits the dependencies between the customer latent space and the agent latent space, which results in generating more correlated customer summaries and agent summaries.

Ablation Study We perform ablations to validate each component of SuTaT by: removing the variational latent spaces (SuTaT w/o LS) so the encoded utterances are directly used for embedding, removing the sentence-level self-attention mechanism (SuTaT w/o Att), and removing the partial copy mechanism (SuTaT w/o copy). We use LSTM encoders and decoders for all ablation models. The results for ablation study in Table 2 show that all the removed components play a role in SuTaT. Removing the latent spaces has the biggest influence on the summarization performance, indicating that the variational latent space is necessary to support our design which makes the agent latent variable dependent on the customer latent variable. The performance drop after removing the sentence-level self-attention mechanism shows that using weighted combined utterance embedding is better than simply taking the mean of encoded utterances. Removing the partial copy has the smallest quality drop. However, taking the dialogue example in Table 1, without the partial copy mechanism SuTaT can generate the following summaries:

Customer Summary: i would like to book a hotel in cambridge on tuesday .

Agent Summary: i have booked you a hotel . the reference number is lzludtvi . can i help you with anything else ?

The generated summaries are the same except for the wrong reference number which is crucial information in this summary.

Classification with Summaries

A good dialogue summary should reflect the key points of the utterances. We perform dialogue classification based on dialogue domains to test the validity of generated summaries. First we encode the generated customer summary and agent summary into $\boldsymbol{e}_X$ and $\boldsymbol{e}_Y$ using the trained encoders of each model, which are then concatenated as features of the dialogue for classification. In this way, the dialogue features are obtained unsupervisedly. Then we train a separate linear classifier on top of the encoded summaries. We use SuTaT with LSTM encoders and decoders for this task. As shown in Table 5, SuTaT outperforms other baselines on dialogue classification, indicating the SuTaT generated summaries have better comprehension of domain information in the dialogue.

Customer:	yes , yes . are there any multiple sports places that i can visit in ?
Agent:	sorry , there are none locations in the center of town . would you like a different area ?
Customer:	yes please . book for the same group of people at 13:45 on thursday .
Agent:	your booking was successful and your reference number is minorhoq .
Customer:	hi , i am looking for a place to stay . the west should be cheap and doesn't need to have internet .
Agent:	there are no hotels in the moderate price range . would you care to expand other criteria ?

Table 6: Examples of single-turn conversations generated by the conditional generative module of SuTaT.

We can also perform supervised classification by using $\boldsymbol{s}_X$ and $\boldsymbol{s}_Y$ from SuTaT as features to train a linear classifier. The cross entropy loss is combined with Equation 4 as the new objective function where all parameters are jointly optimized. As can be seen in Table 5, the supervised classification results are as high as 0.99 on MultiWOZ and 0.96 on Taskmaster, further demonstrating the effectiveness of SuTaT.

Single-Turn Conversation Generation

The design of the conditional generative module in SuTaT enables generating novel single-turn conversations. By sampling the customer latent variable from the standard Gaussian $\boldsymbol{z}_x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then sampling the agent latent variable $\boldsymbol{z}_y \sim p(\boldsymbol{z}_y|\boldsymbol{z}_x)$, SuTaT can produce realistic-looking novel dialogue pairs using the customer decoder and agent decoder. Table 6 shows three examples of novel single-turn conversations generated by SuTaT using randomly sampled latent variables. We can see that the dialogue pairs are closely correlated, meaning the dependencies between two latent spaces are successfully captured.

Conclusion

We propose SuTaT, an unsupervised abstractive dialogue summarization model, accommodating the two-speaker scenario in tete-a-tetes and summarizing them without using any data annotations. The conditional generative module models the customer utterances and agent utterances separately using two encoders and two decoders while retaining their correlations in the variational latent spaces. In the unsupervised summarization module, a sentence-level self-attention mechanism is used to highlight more informative utterances. The summary representations containing key information of the dialogue are decoded using the same decoders from the conditional generative module, with the help of a partial copy mechanism, to generate a customer summary and an agent summary. The experimental results show the superiority of SuTaT for unsupervised dialogue summarization and the capability for more dialogue tasks.

Ethical Impact

This work moves a step further to investigate how to generate abstractive summaries in dialogue systems without using any data annotations. The most direct society impact could be on dialogue systems in contact centers. The model could perform as an assistance in the process of a conversation between a customer and an agent by automatically generating summary notes, which could significantly alleviate the burden of agents and improve the efficiency to address more customers' problems. From the technical perspective, this work redefines abstractive dialogue summarization as summarizing for each speaker, which simplifies the tricky problem so that merging information from different speakers and changing pronouns are no longer needed. For the reality that dialogues datasets do not have large-scale high-quality summary labels, our work provides a solution.

References

Amplayo, R. K.; and Lapata, M. 2020. Unsupervised Opinion Summarization with Noising and Denoising. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Baziotis, C.; Androutsopoulos, I.; Konstas, I.; and Potamianos, A. 2019. SEQˆ 3: Differentiable Sequence-to-Sequence-to-Sequence Autoencoder for Unsupervised Abstractive Sentence Compression. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Jozefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*.

Bražinskas, A.; Lapata, M.; and Titov, I. 2020. Unsupervised Opinion Summarization as Copycat-Review Generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gašić, M. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* .

Byrne, B.; Krishnamoorthi, K.; Sankar, C.; Neelakantan, A.; Duckworth, D.; Yavuz, S.; Goodrich, B.; Dubey, A.; Cedilnik, A.; and Kim, K.-Y. 2019. Taskmaster-1: Toward a Realistic and Diverse Dialog Dataset. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*.

Chen, L.; Zhang, Y.; Zhang, R.; Tao, C.; Gan, Z.; Zhang, H.; Li, B.; Shen, D.; Chen, C.; and Carin, L. 2019. Improving sequence-to-sequence learning via optimal transport. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Chu, E.; and Liu, P. J. 2019. MeanSum: a neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Erkan, G.; and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* .

Fevry, T.; and Phang, J. 2018. Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*.

Galley, M. 2006. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Goo, C.-W.; and Chen, Y.-N. 2018. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *IEEE Spoken Language Technology Workshop (SLT)*.

Graves, A.; Jaitly, N.; and Mohamed, A.-r. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *IEEE workshop on automatic speech recognition and understanding*.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kingma, D. P.; and Welling, M. 2014. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Li, M.; Zhang, L.; Ji, H.; and Radke, R. J. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Li, P.; Wang, Z.; Lam, W.; Ren, Z.; and Bing, L. 2017. Salience estimation via variational auto-encoders for multi-document summarization. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *ACL Workshop on Text Summarization Branches Out*.

Liu, C.; Wang, P.; Xu, J.; Li, Z.; and Ye, J. 2019. Automatic Dialogue Summary Generation for Customer Service. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Mihalcea, R.; and Tarau, P. 2004. Textrank: Bringing order into text. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*.

Pan, H.; Zhou, J.; Zhao, Z.; Liu, Y.; Cai, D.; and Yang, M. 2018. Dial2desc: end-to-end dialogue description generation. *arXiv preprint arXiv:1811.00185* .

Rossiello, G.; Basile, P.; and Semeraro, G. 2017. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing Workshop on Summarization and Summary Evaluation Across Source Types and Genres*.

Schumann, R. 2018. Unsupervised abstractive sentence summarization using length controlled variational autoencoder. *arXiv preprint arXiv:1809.05233* .

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems (NeurIPS)*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*.

Wang, L.; and Cardie, C. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

West, P.; Holtzman, A.; Buys, J.; and Choi, Y. 2019. BottleSum: Unsupervised and Self-supervised Sentence Summarization using the Information Bottleneck Principle. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*.

Xie, S.; Liu, Y.; and Lin, H. 2008. Evaluating the effectiveness of features and sampling in extractive meeting summarization. In *2008 IEEE Spoken Language Technology Workshop (SLT)*.

Yang, Z.; Zhu, C.; Gmyr, R.; Zeng, M.; Huang, X.; and Darve, E. 2020. TED: A Pretrained Unsupervised Summarization Model with Theme Modeling and Denoising. *arXiv preprint arXiv:2001.00725* .

Yuan, L.; and Yu, Z. 2019. Abstractive Dialog Summarization with Semantic Scaffolds. *arXiv preprint arXiv:1910.00825* .

Zhang, X.; Li, Y.; Shen, D.; and Carin, L. 2018. Diffusion maps for textual network embedding. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhang, X.; Yang, Y.; Yuan, S.; Shen, D.; and Carin, L. 2019. Syntax-Infused Variational Autoencoder for Text Generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zheng, H.; and Lapata, M. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.