Unsupervised Summarization for Chat Logs with Topic-Oriented Ranking and Context-Aware Auto-Encoders

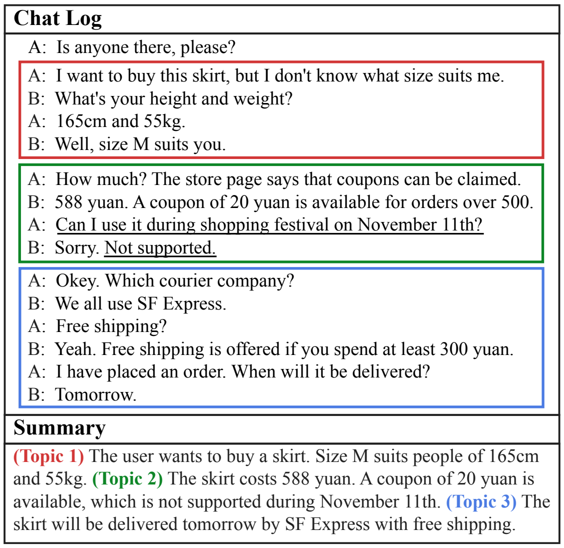
**Yicheng Zou,[1] Jun Lin,[2] Lujun Zhao,[2] Yangyang Kang,[2] Zhuoren Jiang,[3]
Changlong Sun,[3,2] Qi Zhang,[1] Xuanjing Huang,[1] Xiaozhong Liu[4]**
[1]School of Computer Science, Fudan University, Shanghai, China
[2]Alibaba Group, China
[3]Zhejiang University, Hangzhou, China
[4]Indiana University Bloomington, Bloomington, United States
{yczou18, qz, xjhuang}@fudan.edu.cn, {linjun.lj, lujun.zlj, yangyang.kangyy}@alibaba-inc.com,
jiangzhuoren@zju.edu.cn, changlong.scl@taobao.com, liu237@indiana.edu

Abstract

Automatic chat summarization can help people quickly grasp important information from numerous chat messages. Unlike conventional documents, chat logs usually have fragmented and evolving topics. In addition, these logs contain a quantity of elliptical and interrogative sentences, which make the chat summarization highly context dependent. In this work, we propose a novel unsupervised framework called *RankAE* to perform chat summarization without employing manually labeled data. *RankAE* consists of a topic-oriented ranking strategy that selects topic utterances according to centrality and diversity simultaneously, as well as a denoising auto-encoder that is carefully designed to generate succinct but context-informative summaries based on the selected utterances. To evaluate the proposed method, we collect a large-scale dataset of chat logs from a customer service environment and build an annotated set only for model evaluation. Experimental results show that *RankAE* significantly outperforms other unsupervised methods and is able to generate high-quality summaries in terms of relevance and topic coverage.

Introduction

The goal of text summarization is to generate a succinct summary while retaining a document's essential information. From a participation viewpoint, most existing works focus on single-party documents like news, reviews, and scientific articles (See et al. 2017; Nikolov et al. 2018; Narayan et al. 2018; Chu and Liu 2019). Meanwhile, multi-party chat conversations are generated online every day but have not been fully explored. Despite the considerable research on similar dialogues, like meetings and telephone records (Zechner 2001; Gurevych and Strube 2004; Gillick et al. 2009; Shang et al. 2018), chat summarization has its own characteristics. Compared with other dialogue forms, chat logs are generally pure text without audio or transcription information and tend to be much shorter, more unstructured, and contain more spelling mistakes, hyperlinks, and acronyms (Uthus and Aha 2011; Koto 2016).



Chat Log

A: Is anyone there, please?

A: I want to buy this skirt, but I don't know what size suits me.
B: What's your height and weight?
A: 165cm and 55kg.
B: Well, size M suits you.

A: How much? The store page says that coupons can be claimed.
B: 588 yuan. A coupon of 20 yuan is available for orders over 500.
A: Can I use it during shopping festival on November 11th?
B: Sorry. Not supported.

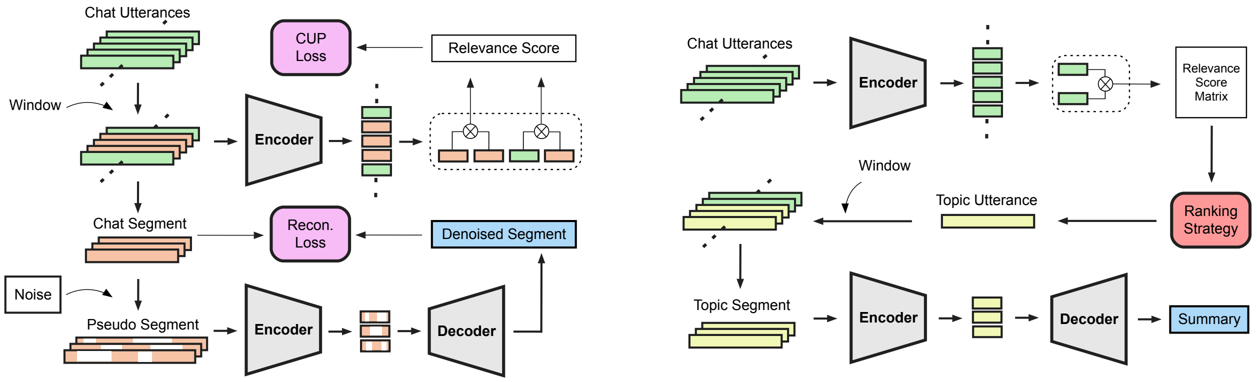
A: Okey. Which courier company?
B: We all use SF Express.
A: Free shipping?
B: Yeah. Free shipping is offered if you spend at least 300 yuan.
A: I have placed an order. When will it be delivered?
B: Tomorrow.

Summary

(Topic 1) The user wants to buy a skirt. Size M suits people of 165cm and 55kg. **(Topic 2)** The skirt costs 588 yuan. A coupon of 20 yuan is available, which is not supported during November 11th. **(Topic 3)** The skirt will be delivered tomorrow by SF Express with free shipping.

Figure 1: Example of a chat log and its reference summary. It has three topics: *Skirt Size*, *Price* and *Logistics*. Utterances in the same color box describe the same topic. An interrogative sentence with its elliptical response is underlined.

Most existing summarization works on conventional documents aim to extract salient sentences or form an abstractive expression that captures the main idea of a document (See et al. 2017; Narayan et al. 2018; Liu and Lapata 2019). Nevertheless, in chat conversations, chat topics can be diverse and switch frequently as the conversation progresses, namely the '*Topic Shift*' (Arguello and Rosé 2006; Sood et al. 2013) phenomenon. Figure 1 shows a chat log example with three topics that have different user intentions and text semantics. To address this problem, most previous works have conducted clustering (Zhou and Hovy 2005), chat segmentation (Sood et al. 2012), and fine-grained topic modeling (Sood et al. 2013) to extract the utterances with different semantics. However, chat logs always contain numerous elliptical and interrogative sentences that are highly dependent on their context. As shown in Figure 1, questions and



(a) The training stage of RankAE.

(b) The inference stage of RankAE.

Figure 2: The overall framework of RankAE. (a) Chat segments are composed of utterances in a specific window scope[1]. CUP denotes the Context Utterance Prediction that produces the co-occurrence probability of two utterances to measure the relevance score for utterance ranking. Original chat segments are extended with noisy content and then recovered by training the DAE. (b) At inference time, the model first selects topic utterances by the extractive module and then filters out noisy information from corresponding topic segments to perform segment compression for generating concise summaries.

responses like ’How much?’ and ’Not supported.’ could be meaningless if the necessary context information is missing. This linguistic phenomenon requires systems to fuse context information and produce integral descriptions. Obviously, general extractive approaches are not the ideal solution to address the problem. On the other hand, abstractive approaches for text and dialogue summarization (See et al. 2017; Narayan et al. 2018; Liu et al. 2019a) may be promising for context information integration and refinement. Most of those approaches share a similar prerequisite: *a large decent training dataset with annotated summaries*. However, existing datasets for chat summarization are still very limited due to the expensive labeling cost, which makes the supervised abstractive methods difficult to apply.

To tackle the topic shift problem in chat logs and the information integrity problem of individual utterances, in this work we introduce a novel unsupervised neural framework called *RankAE* that benefits from both extractive and abstractive paradigms. **First**, we propose a novel ranking strategy to identify *topic utterances* (the utterances that express distinct topics and semantics), under the assumption that utterances describing the same topic tend to be located near to each other (Passonneau and Litman 1993). Topic utterances are selected by running a diversity-enhanced ranking algorithm based on the co-occurrence probability of each utterance pair in a specific context scope. **Second**, for each utterance, we collect the surrounding utterances to form a *chat segment* that captures contextual information. However, original chat segments may contain irrelevant and redundant content. Hence, we further leverage a denoising auto-encoder (DAE) (Vincent et al. 2008) and modify its training regime to perform segment compression. The overall network can be trained end-to-end. At the inference stage, our model can select topic utterances and then generate condensed but context-informative summaries by compressing their corresponding chat segments. In this work, we further collected a large-scale chat log dataset from an e-commerce platform, along with a small annotated subset only for evaluation. Experiments on the dataset showed that *RankAE* outperformed other unsupervised methods under different evaluation metrics. Codes and datasets are publicly available[2].

In summary, our contributions are three-fold: 1) We propose a novel neural framework for chat log summarization in a fully unsupervised manner. 2) The framework benefits from both extractive and abstractive paradigms, which can not only capture critical and topic-diverse information but also generate succinct and context-aware summaries. 3) Comprehensive studies on a large real-world chat log dataset show the effectiveness of our method in different aspects.

Proposed Method

The RankAE has two components: a topic utterance extractor and a denoising auto-encoder (DAE) (Vincent et al. 2008). For each utterance, we collect its surrounding utterances to form a chat segment. At training time, the extractor learns to predict the relevance score for each utterance pair. Meanwhile, chat segments are extended with noisy content and then recovered by training the DAE generator. At inference time, topic utterances are selected by running a diversity-enhanced ranking algorithm based on the relevance scores. Then, all topic segments (the chat segment of topic utterance) are compressed with the auto-encoder by filtering out nonessential information. The compressed segments are concatenated to form the final summary. The overall training and inference stages are illustrated in Figure 2.

End-to-End Training Stage

BERT Encoder & Multi-Party Information. In this work, we use BERT (Devlin et al. 2019) as the utterance encoder

[1] In Figure 2, each chat segment is composed of a central utterance u_i and two adjacent utterances u_{i-1}, u_{i+1}.

[2] https://github.com/RowitZou/RankAE

for RankAE, which is a powerful encoder pre-trained on large-scale corpora. We denote each chat log as an utterance sequence $D = \{u_1, u_2, ..., u_n\}$. To incorporate multi-party information, for the i-th utterance in a chat log, we have $u_i = \{p_i, w_{i1}, .., w_{im}\}$, where p_i is an embedding representing the current party of u_i, and w_{ij} is the embedding of the j-th word. Each utterance u_i is encoded by BERT, and the utterance representation h_i is derived from the output vector of the first token ([CLS] token) at the last layer:

$$h_i = \mathrm{BERT}(u_i). \qquad (1)$$

Context Utterance Prediction. To encourage BERT to better understand utterance relationships, inspired by the sentence distributional hypothesis (Zheng and Lapata 2019), we design an utterance-level training objective called Context Utterance Prediction (CUP) to classify whether two utterances are near in the context. For each utterance u_i, we collect its surrounding utterances with a window size c to form a *chat segment*, formally $S_i = \{u_{i-c}, ..., u_i, ..., u_{i+c}\}$. All utterances in S_i are positive examples, while others are negative examples. Similar to Mikolov at al. (2013), we employ negative sampling and define the CUP loss as follows:

$$\mathcal{L}_{cup} = \sum_{-c\leq j\leq c, j\neq 0} \log\sigma(h_{u_{i+j}}^{\top} W h_{u_i}) + \sum_{j=1}^{m} \mathbb{E}_{u_j \sim \mathrm{P}(u)}[\log\sigma(-h_{u_j}^{\top} W h_{u_i})], \qquad (2)$$

where $\mathrm{P}(u_i, u_j) = \sigma(h_{u_j}^{\top} W h_{u_i})$ represents the utterance co-occurrence probability in a specific context scope, which can measure the relevance of two utterances. $\mathrm{P}(u)$ is a uniform distribution from which we sample m negative examples for each positive data point. W is a trainable parameter[3]. Notably, unlike the original pre-training task for BERT, we sample negative examples from the same chat log instead of from the whole corpus, which is more challenging as utterances in the same chat log are more similar and confusing.

Noise Addition. Compared to individual utterances, chat segments are context-informative but may contain irrelevant and redundant content. To tackle this problem, we employ the Denoising Auto-Encoder (DAE) (Vincent et al. 2008) to perform text compression. Given an original chat segment, we extend it with noise to construct a pseudo segment. The modified segments and original ones compose training pairs. The model is then trained to exclude noisy information and recover original segments. Since we employ the BPE tokenization mechanism in BERT, words with spelling errors and acronyms will be tokenized into inappropriate subtokens. Thus, we could transfer character-level typos into token-level errors. Accordingly, for each utterance in a chat segment, we design the following modification procedure to add multi-granularity noise, which has three options:

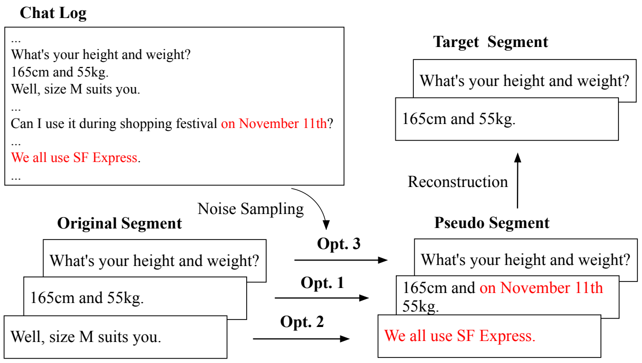


Figure 3: Noise addition for a chat segment. Texts with red color represent noise sampled from the same chat log. Opt.1, Opt.2 and Opt.3 represent fragments insertion, utterance replacement, and content retention, respectively. The replaced utterance in Opt.2 is also removed from the target segment.

- Inspired by Fevry and Phang (2018), we randomly sample utterances from the same chat log and sub-sample some word spans as noisy fragments, which are inserted in the original utterance until the length of the sequence is increased by a ratio of 40% to 60%. This option called **fragments insertion** is performed with probability p_a.
- With probability p_r, the whole utterance is replaced with another one in the same chat log, namely **utterance replacement**. Accordingly, the replaced utterance is also removed from the training target so that our model learns to filter out irrelevant utterances on a coarse-grained level.
- Keep the utterance unchanged with probability p_s. The purpose of **content retention** is to bias the representation towards the actual observed utterance.

Here, $p_a + p_r + p_s = 1$. An example of noise addition for chat segments is shown in Figure 3.

Chat Segment Reconstruction. After noise addition, we obtain the pseudo chat segment of central utterance u_i, denoted as $\widetilde{S}_i = \{\tilde{u}_{i-c}, ..., \tilde{u}_i, ..., \tilde{u}_{i+c}\}$. All utterances in $\widetilde{S}_i$ are also encoded by BERT as in Eq.1 where the BERT parameter weights are shared. The output representations are $\widetilde{H}_i = [\tilde{h}_{i-c}, ..., \tilde{h}_i, ..., \tilde{h}_{i+c}]$. However, directly training the DAE to recover original segments is unstable, as it may discard information randomly without a conditional guidance. Hence, we take u_i as a query to match relevant content in $\widetilde{S}_i$. Here, we use a Transformer Encoder (Vaswani et al. 2017) to capture chat semantics and form queries q_i as follows:

$$[q_1, q_2, ..., q_n] = \mathrm{TransEnc}([h_1, h_2, ..., h_n]). \qquad (3)$$

The decoder is also implemented by the Transformer with a masked attention mechanism for auto-regressive generation:

$$p(\hat{S}_i) = \mathrm{TransDec}(\widetilde{H}_i; q_i). \qquad (4)$$

$\hat{S}_i$ is the predicted chat segment. q_i acts as a beginning-of-sequence input embedding in the decoding process to control the generation results. $\widetilde{H}_i$ is the encoder memory. Notably, our decoder applies utterance representations as memories instead of using word-level attention or copy mechanism. It encourages all semantics to be captured in $\widetilde{H}_i$. Finally, we use the original segment S_i as a gold reference to

[3] Here we can also employ the dot product without the parameter matrix W similar to Zheng and Lapata (2019), but we found it empirically more effective to add extra parameters.

train the auto-encoder for chat segment reconstruction:

$$\mathcal{L}_{rec} = -\sum\nolimits_{i} \text{log}p(\hat{S}_i). \tag{5}$$

Joint Training. Finally, we combine two loss functions in Eq.2 and Eq.5 to jointly train the model, where α is a hyper-parameter to adjust the loss proportion:

$$\mathcal{L} = \alpha\mathcal{L}_{cup} + (1-\alpha)\mathcal{L}_{rec}. \tag{6}$$

Utterance Ranking and Summary Generation

Topic Utterance Selection. At the inference stage, given the utterance representations $H = [h_1, h_2, ..., h_n]$ derived from Eq.1, we can obtain the utterance relevance matrix as:

$$\widetilde{\mathrm{M}}_{ij} = \sigma(h_j^\top W h_i). \tag{7}$$

Each score $\widetilde{\mathrm{M}}_{ij}$ is calculated as the utterance co-occurrence probability in Eq.2 to measure relevance between utterances. Moreover, under the assumption that utterances describing the same topic tend to appear in near contexts, we add a distance coefficient λ^L to constrain the score of distant utterance pairs. According to the Gaussian distribution, we can get λ^L and the final relevance matrix M as follows:

$$\lambda_{ij}^L = \exp[-\frac{(\mathrm{P}_j - \mathrm{P}_i)^2}{2(n/k)^2}], \tag{8}$$

$$\mathrm{M}_{ij} = \lambda_{ij}^L \, \widetilde{\mathrm{M}}_{ij}, \tag{9}$$

where $1 \leq \mathrm{P}_i, \mathrm{P}_j \leq n$ represents the absolute position of utterance u_i, u_j in a chat log. n denotes the utterance number and k represents the expected number of topic utterances. M can further be regarded as an adjacent matrix of an undirected graph, and the centrality degree for utterance u_i is calculated as follows similar to Erkan and Radev (2004):

$$C(u_i) = \sum_{1 \leq j \leq n, j \neq i} \mathrm{M}_{ij}, \tag{10}$$

which can be called *local centrality score* since λ^L highlights the local contexts of u_i. $C(u_i)$ is a score for ranking algorithms to select the best utterance candidates. However, it only takes centrality into account and ignores the topic diversity among selected utterances. Inspired by Maximal Marginal Relevance (MMR) (Carbonell and Goldstein 1998), we modify Eq.10 to produce a score that satisfies both quality and topic diversity. Specifically, we define R as a set which includes all utterances in a chat log and define V as the current topic utterances set. For $u_i \in R - V$, we have:

$$C(u_i) = \frac{\eta}{n-1} \sum_{u_j \in R, j \neq i} \mathrm{M}_{ij} - (1-\eta) \max_{u_j \in V} \mathrm{M}_{ij}, \tag{11}$$

where η is a coefficient in the range $[0, 1]$ to adjust the preference of relevance or diversity. At the beginning of ranking algorithm, V is an empty set. At each iteration step, the utterance with maximum $C(u_i)$ in $R - V$ will be added to V until the number of topic utterances exceeds k:

$$V \leftarrow \arg \max_{u_i \in R - V} C(u_i). \tag{12}$$

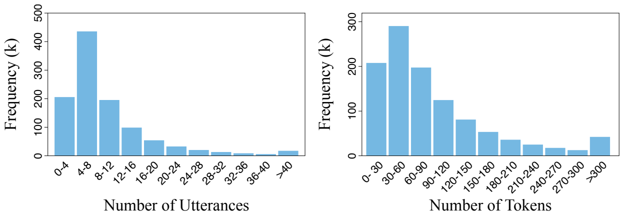


Figure 4: The left figure shows the distribution of utterance number in chat logs. The right figure shows the distribution of chat log length (number of tokens in a chat log).

Summary Generation. After selecting topic utterances, we can simply combine them to create a summary. However, it may miss out on critical information without their contexts. Hence, topic segments are constructed based on the selected topic utterances. Then, we input these topic segments into DAE without any modification, expecting the model to further filter out nonessential content. Formally, for $u_i \in V$, we input utterance representations H_i, namely $H_i = [h_{i-c}, ..., h_i, ..., h_{i+c}]$, and the query embedding q_i into the Transformer decoder as follows:

$$p(\hat{S}_i) = \mathrm{TransDec}(H_i; q_i). \tag{13}$$

The compressed topic segment $\hat{S}_i$ is then decoded by using a beam search algorithm just like other abstractive summarization works (See et al. 2017). The final summary is created by concatenating all the condensed segments.

Experimental Settings

Dataset

Our chat log dataset is collected from an E-commerce platform where conversations take place between customers and merchants in the Chinese language. The dataset contains 1.09M chat logs and 10.03M utterances. To process the raw data, we replaced specific information like numbers, URLs, and e-mail addresses with special tokens. In addition, some filler words in chat logs and common words in customer service scenarios like ’um’, ’well’, and ’hello’ were also discarded. Statistics of the processed dataset are shown in Figure 4. The average utterance number in a chat log is 9.22. The average length (number of tokens) of utterances and chats is 10.30 and 94.90, respectively.

To perform model evaluation, we further randomly sampled 1000 chat log examples for summary annotation, including 500 validation examples and 500 test examples. The remainder of the chat logs were unlabeled and treated as training data. All gold summaries were annotated by three experienced and independent experts under a uniform criterion. Specifically, we first extracted topic points in each chat, such as *price* or *logistics*. Then, topic points were expanded into succinct and complete sentences as sub-summaries, which describe the main ideas of topic points conveyed by the original chat. These sub-summaries were concatenated as the final summary. The average length (number of tokens) of gold summaries is 36.56. A chat log example and its reference summary can be found in Figure 1.

Comparison Methods

We applied several comparison methods for chat summarization, which were all designed in unsupervised scenarios. **Lead** (Nallapati et al. 2017) simply extracts the first several sentences in a document as the summary, which can represent the lower bound of extractive methods. **Oracle** (Nallapati et al. 2017) uses a greedy algorithm to select the best performing sentences against the gold summary. It represents the upper bound of extractive methods. **TextRank** (Mihalcea and Tarau 2004) converts documents into graphs and selects sentences by running a graph-based ranking algorithm. **MMR** (Carbonell and Goldstein 1998) extracts sentences considering both relevance and diversity. **PacSum** (Zheng and Lapata 2019) improves TextRank by building directed graphs and adding weight constraints based on the edge direction. **MeanSum** (Chu and Liu 2019) uses the mean of the representations from an auto-encoder for input sentences to decode a summary. **SummAE** (Liu et al. 2019b) generates short summaries by jointly reconstructing documents and sentences using a DAE and an adversarial critic. In our experiments, we also evaluated its non-critic variant.

Hyper-parameter Settings

We used the pre-trained Chinese BERT model released by Cui et al. (2019). To alleviate the mismatch between the pre-trained BERT and other randomly initialized parameters, we used different optimizers similar to Liu et al. (2019). Specifically, we employed Adam (Kingma and Ba 2014) with learning rate 1e-3 for BERT and 1e-2 for other parameters. Our Transformer layers has 768 hidden units, 8 heads, and the hidden size for all feed-forward layers is 2,048. The model was trained for 200,000 steps with 10,000 warm-up steps on a Tesla V100 32GB GPU. Model checkpoints were saved and evaluated on the validation set every 2,000 steps. Checkpoints with top-3 performance were finally evaluated on the test set to report averaged results. We truncated each chat to 40 utterances[4]. Utterances with more than 40 tokens were also truncated[5]. For the negative sampling in Eq.2, we sampled two negative examples for each positive data point. The loss coefficient α and the relevance-diversity coefficient η were set to 0.5 for a balanced choice. Compared to spoken dialogues like meetings, chat logs are much shorter (about 91% of chats have up to 20 utterances), so that a window size $c = 1$ is sufficient and we set $k = n/(2c+1)$ with an upper bound 3. As for $\{p_a, p_r, p_s\}$, a high value of p_r leads to an over-noise pseudo segment, while a high value of p_s makes the noise addition insufficient. As a result, we set $\{p_a, p_r, p_s\}$ to $\{0.7, 0.2, 0.1\}$. In this setting, given $c = 1$ (up to three utterances in each segment), the probability of replacing at least one utterance is $1 - \mathtt{C}_3^0 p_r^0 (1-p_r)^3 = 0.488$, according to the multinomial distribution.

Results and Analysis

In this section, we show the results of RankAE and other unsupervised methods for chat summarization. We also probe the effectiveness of RankAE by explanatory experiments.

Table 1: Results on the E-commerce chat log dataset. Methods are categorized into three groups: baseline, extractive and abstractive methods. TRF denotes the Transformer.

Methods	**RG-1**	**RG-2**	**RG-L**	**BLEU**
LEAD	19.32	10.78	19.12	12.47
ORACLE	47.18	27.89	45.96	29.53
TextRank / tf-idf	22.34	11.22	21.49	15.33
TextRank / BERT	22.16	11.34	21.77	15.40
PacSum / tf-idf	21.87	10.82	21.01	15.16
PacSum / BERT	22.10	11.05	21.33	15.24
MMR / tf-idf	23.75	12.11	22.94	15.57
MMR / BERT	23.92	12.27	23.06	15.76
RankAE (Ext.) / tf-idf	24.52	12.49	23.61	15.92
RankAE (Ext.) / BERT	**25.10**	**12.60**	**23.92**	**16.13**
MeanSum / RNN	18.66	8.39	18.13	10.91
MeanSum / TRF	19.04	8.92	18.57	11.00
SummAE / RNN	19.29	9.21	18.87	11.43
SummAE / TRF	20.37	9.81	19.86	11.65
SummAE - critic / RNN	25.30	12.62	24.75	14.02
SummAE - critic / TRF	26.17	13.58	25.63	14.29
RankAE - BERT	27.63	14.32	27.14	16.66
RankAE	**28.20**	**14.76**	**27.59**	**16.87**

Main Results

We use ROUGE (Lin 2004) and BLEU (Papineni et al. 2002) to evaluate the methods. We report ROUGE-1 (RG-1), ROUGE-2 (RG-2) and ROUGE-L (RG-L) F-scores that measure the unigram, bigram and longest common sequence overlaps between the references and prediction summaries. BLEU measures the n-gram precision, where we report averaged scores with 4-grams at most in our experiments.

Table 1 shows the main results of RankAE and other comparison methods. Summaries from all systems are constrained to a maximum length of 40 tokens[6] for a fair comparison. The first part in Table 1 includes LEAD and ORACLE baselines. The second part is extractive methods, where we experiment with two utterance representations to compute the score matrix M. The first one is based on tf-idf similar to Zheng and Lapata (2019). Cosine similarity scores are calculated for these tf-idf vectors to build the score matrix. The second one is based on BERT with the same fine-tuning process as proposed in Eq.2. We use the relevance scores computed in Eq.7 to form the score matrix. RankAE(Ext.) stands for the topic utterance extractor, where the selected topic utterances are directly concatenated as the final summary without adding context. For abstractive methods in the third part, we use BERT as the utterance encoder except for RankAE-BERT, which is a variant of RankAE and leverages the basic Transformer encoder without pre-training. On the other hand, RankAE employs the Transformer decoder for generation, while other baselines originally use RNN (Schuster and Paliwal 1997). For a fair comparison, we also implement the Transformer decoder for these methods.

Results in Table 1 show that RankAE(Ext.) achieves competitive performances against other extractive methods on

[4] About 1.60% of chats were truncated.

[5] About 3.12% of utterances were truncated.

[6] Output summaries exceeding 40 tokens are truncated. The average length of gold summaries is 36.56.

Table 2: Ablation Study. The first part includes variants of the extractor in RankAE. The second part shows the results under different settings of DAE. L-Rto. denotes the length ratio between system summaries and gold references. c denotes the window size.

Models	L-Rto.	RG-1	RG-L	BLEU
RankAE (Ext.)	0.96	27.19	25.32	17.31
- distance constraint	0.95	25.68	24.52	16.64
- diversity enhancement	0.95	24.11	23.39	16.31
+ context ($c = 1$)	2.23	34.66	34.20	16.92
+ context ($c = 2$)	2.71	34.82	34.17	16.75
RankAE (w/o noise add.)	2.21	34.53	34.07	16.90
+ noise add. (20%)	1.60	32.34	31.60	17.28
+ noise add. (40%)	1.15	30.49	29.84	17.62
+ noise add. (60%)	0.97	30.30	29.77	17.85

all metrics, which validates the effectiveness of the topic-oriented ranking strategy for chat summarization. Compared to RankAE(Ext.), the full framework with DAE generator improves the results by a large margin (+2.53, +1.72, +3.22 on ROUGE-1/2/L). It manifests that, beyond the extractive paradigm, our model is capable of integrating context information and generating summaries that are more relevant to original chat logs. When equipped with BERT, RankAE gives a further improvement. The results of RankAE have a statistically significant difference from all other methods (except RankAE-BERT) using a Wilcoxon signed-rank test with p <0.05, which verifies the effectiveness of RankAE that benefits from both extractive and abstractive paradigms.

Ablation Study

We also perform ablation studies to validate the effectiveness of each part in RankAE. Table 2 demonstrates the results of different settings for the proposed framework equipped with BERT. In order to show the impact of noise addition and text compression on summary lengths, we calculate the averaged length ratio between output summaries $\hat{S}$ and gold references S without summary truncation[7]:

$$\text{Length Rto.} = \frac{\text{Length}(\hat{S})}{\text{Length}(S)}. \qquad (14)$$

After removing the distance constraint λ^L or the diversity enhancement mechanism in Eq.11, we observe a performance degradation, which verifies that the topic diversity and the utterance distance are two important factors that influence the results of utterance extraction for chat logs. It indicates that chat logs may have a wide topic coverage with different semantics, while utterances describing the same topic usually locate near to each other. After collecting context utterances in the chat segment, the ROUGE score is unsurprisingly boosted where the average length is about twice longer than gold summaries. However, the BLEU score decreases, which possibly means redundant content is also collected. Besides, a larger window size (c=2) does not necessarily mean a better performance. One possible factor is that chat logs are much shorter than regular documents, and a small window size might be enough to cover sufficient information. The second part of Table 2 shows the influence of noise addition, where the percentage means the ratio of utterance extension by adding noise in the process of fragments insertion. Results show that the summary length is effectively restricted as the ratio of noise addition increases because a higher ratio requires the DAE to filter out more information. Notably, compared to RankAE(Ext.), RankAE with a 60% of noise addition achieves better results while the average length of their summaries nearly have no difference. It shows that although RankAE integrates more context information, it is still capable of excluding irrelevant and redundant content and generating short summaries under the premise of a high performance.

Table 3: Human evaluation results in relevance and succinctness. The score represents the percentage of times each method is chosen as better in pairwise comparisons.

Models	Relevance	Succinctness
TextRank	37.8%	41.7%
SummAE	28.2%	45.7%
RankAE (Ext.)	39.3%	**50.4%**
RankAE	**57.2%**	46.4%
Gold	87.5%	65.8%

Human Evaluation

Considering automatic metrics like ROUGE and BLEU may not suitably represent the content to be evaluated, we randomly sample 100 cases in the test set and invite volunteers to evaluate the summaries. The process of human evaluation is designed similar to Narayan at al. (2018). Specifically, volunteers are presented with one chat and two summaries produced from two different systems and are asked to decide which summary is better in terms of two dimensions: relevance (which summary captures more information relevant to the original chat?) and succinctness (which summary contains fewer redundant content?). In order to minimize the inter-human noise, we collect judgments from three volunteers for each comparison. We also randomize the order of summaries and chats for each volunteer.

We compare our model RankAE against TextRank, SummAE, RankAE(Ext.), and the human reference (Gold) (see Table 3). The score of each method is the percentage of times it is chosen as better given 2 summaries from 2 out of 5 systems. Unsurprisingly, gold summaries are considered better in most of the time. In terms of relevance, RankAE outperforms other comparison methods significantly, indicating that RankAE can generate more relevant summaries covering different topics and corresponding contexts. In terms of succinctness, RankAE(Ext.) produces more summaries ac-

[7] Without summary truncation, the scores in Table 2 might be slightly higher than those in Table 1.

Table 4: An example of chat summarization with RankAE. Texts with red color represent nonessential or redundant content in the chat segment, which are excluded by RankAE to produce a more concise summary.

Chat Log	A: Dear, I am at your service online. B: How much is it? B: Please quote a price inclusive of shipping. A: 180, excluding tax and shipping. B: Would it be my turn to get shipped tomorrow? A: Not sure. Place the order earlier and get shipped earlier.
Gold	The price is 180 excluding tax and shipping. It may not be shipped tomorrow.
RankAE (Ext.)	180, excluding tax and shipping. Not sure. Place the order earlier and get shipped earlier.
RankAE (Ext.) + context	Please quote a price inclusive of shipping. 180, excluding tax and shipping. Would it be my turn to get shipped tomorrow? Would it be my turn to get shipped tomorrow? Not sure. Place the order earlier and get shipped earlier.
RankAE	Please quote a price. 180, excluding tax and shipping. Would it be shipped tomorrow? Not sure.

cepted by volunteers, which means redundancy is effectively reduced with diversity enhancement. When context is incorporated, succinctness might decrease, but RankAE still improves the summary relevance under the premise of redundancy restriction. We also carry out pairwise comparisons between models (using a Binomial Two-Tailed test; null hypothesis: the models are equally good; $p < 0.01$). Gold is significantly different from all other methods. In terms of relevance, RankAE and SummAE are significantly different from other methods. In terms of succinctness, RankAE(Ext.) is significantly different from TextRank. All other model differences are not statistically significant.

Case Study

Table 4 shows an example that probes the ability of RankAE to extract topic utterances and generate concise and context-informative summaries, which is translated from Chinese. The chat has two topics, namely *price* and *shipping issues*. RankAE(Ext.) successfully picks out two topic-relevant utterances. However, some vital information is missed out, such as 'price' and 'tomorrow'. By collecting contexts in the chat segment, the necessary information is supplemented, but nonessential phrases and duplicate utterances are also included, which are marked with red color. Equipped with DAE, RankAE is able to filter out these useless content and finally produces a short and integral summary.

Related Work

Unsupervised Text Summarization

In the task of text summarization, large-scale training data is not always available. As a result, the unsupervised fashion has recently attracted increasing research interest. A couple of works proposed extractive methods for unsupervised summarization, which generally assign salient scores to sentences in a document and select the top-ranked ones to form the summary. Typical methods are based on word frequency (Nenkova and Vanderwende 2005), topic modeling (Harabagiu and Lacatusu 2005), cluster centroid (Radev et al. 2004; Rossiello et al. 2017), sentence graph (Erkan and Radev 2004; Zheng and Lapata 2019), Integer Linear Programming (ILP) optimization (McDonald 2007; Gillick et al. 2009), and sparse coding (He et al. 2012; Liu et al. 2015). Recently, abstractive approaches have been proposed due to the success of deep neural models, where the auto-encoder framework has been applied (Miao and Blunsom 2016; Fevry and Phang 2018; Chu and Liu 2019; Liu et al. 2019b). In this work, we employ both extractive and abstractive paradigms, where a topic-oriented ranking mechanism and a context-aware auto-encoder are combined to stack additional improvements to unsupervised summarization.

Summarization on Chat Logs

Summarization on conversations is a valuable but challenging task that receives much attention in recent years. Most previous works focus on spoken dialogues like telephone records (Zechner 2001; Gurevych and Strube 2004) and meetings (Xie et al. 2008; Mehdad et al. 2013; Shang et al. 2018), which are originally in form of audio and transcribed into texts. Another line of works focus on email threads (Rambow et al. 2004; Murray and Carenini 2008), which is a type of text media similar to chat logs. However, most of them leverage features specific to emails such as mail structures that are not applicable to other text-based conversations. In terms of chat summarization, the most relevant work has been done by Zhou and Hovy (2005) that aims to produce chat summaries comparable to the human made GNUe Traffic digest. Based on the GNUe dataset, some approaches have been explored (Sood et al. 2012, 2013; Mehdad et al. 2014). However, they depend on well-designed feature engineering or external resources, such as special terms from GUNe IRCs or query terms from WordNet synonyms. Moreover, chats in the GNUs dataset are special discussions about technical problems, which are quite different from daily chats. Recently, Koto (2016) proposes another chat log dataset in the Indonesian language. However, both the two datasets only contain a limited number of chats. By contrast, in this work, we collect a large-scale chat log corpus along with a small subset with gold summaries for evaluation. We also propose a fully unsupervised neural framework that can be trained in an end-to-end manner.

Conclusion and Future Work

In this work, we propose a novel unsupervised framework for chat summarization. A topic-oriented ranking strategy is designed to pick out utterances based on local centrality and topic diversity, while a denoising auto-encoder captures context information and discards nonessential content to produce succinct summaries. Future directions may be the topic variation within an utterance, where a more fine-grained ranking strategy on the word level can be explored. We could also consider other ways of noise addition, e.g., deletion and repeating, to introduce more kinds of noise.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key R&D Program (No. 2018YFC0831105), National Natural Science Foundation of China (No. 61751201, 62076069, 61976056), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), Science and Technology Commission of Shanghai Municipality Grant (No.18DZ1201000, 17JC1420200). This work was supported by Alibaba Group through Alibaba Innovative Research Program.

References

Arguello, J.; and Rosé, C. 2006. Topic-segmentation of dialogue. In *Proceedings of the Analyzing Conversations in Text and Speech*, 42–49.

Carbonell, J.; and Goldstein, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 335–336.

Chu, E.; and Liu, P. 2019. MeanSum: A Neural Model for Unsupervised Multi-Document Abstractive Summarization. In *International Conference on Machine Learning*, 1223–1232.

Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; and Hu, G. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101* .

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Erkan, G.; and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22: 457–479.

Fevry, T.; and Phang, J. 2018. Unsupervised Sentence Compression using Denoising Auto-Encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 413–422.

Gillick, D.; Riedhammer, K.; Favre, B.; and Hakkani-Tur, D. 2009. A global optimization framework for meeting summarization. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 4769–4772. IEEE.

Gurevych, I.; and Strube, M. 2004. Semantic similarity applied to spoken dialogue summarization. In *Proceedings of the 20th international conference on Computational Linguistics*, 764. Association for Computational Linguistics.

Harabagiu, S.; and Lacatusu, F. 2005. Topic themes for multi-document summarization. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 202–209.

He, Z.; Chen, C.; Bu, J.; Wang, C.; Zhang, L.; Cai, D.; and He, X. 2012. Document summarization based on data reconstruction. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Koto, F. 2016. A publicly available indonesian corpora for automatic abstractive and extractive chat summarization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 801–805.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Liu, C.; Wang, P.; Xu, J.; Li, Z.; and Ye, J. 2019a. Automatic Dialogue Summary Generation for Customer Service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Liu, H.; Yu, H.; Deng, Z.-H.; et al. 2015. Multi-document summarization based on two-level sparse representation model. In *Twenty-ninth AAAI conference on artificial intelligence*.

Liu, P. J.; Chung, Y.-A.; Ren, J.; et al. 2019b. SummAE: Zero-shot abstractive text summarization using length-agnostic auto-encoders. *arXiv preprint arXiv:1910.00998* .

Liu, Y.; and Lapata, M. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3721–3731.

McDonald, R. 2007. A study of global inference algorithms in multi-document summarization. In *European Conference on Information Retrieval*. Springer.

Mehdad, Y.; Carenini, G.; Ng, R.; et al. 2014. Abstractive summarization of spoken and written conversations based on phrasal queries. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1220–1230.

Mehdad, Y.; Carenini, G.; Tompa, F.; and Ng, R. 2013. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*, 136–146.

Miao, Y.; and Blunsom, P. 2016. Language as a Latent Variable: Discrete Generative Models for Sentence Compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 319–328.

Mihalcea, R.; and Tarau, P. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 404–411.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Murray, G.; and Carenini, G. 2008. Summarizing spoken and written conversations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 773–782.

Nallapati, R.; Zhai, F.; Zhou, B.; et al. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Narayan, S.; Cohen, S. B.; Lapata, M.; et al. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1797–1807.

Nenkova, A.; and Vanderwende, L. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep.* 101.

Nikolov, N. I.; Pfeiffer, M.; Hahnloser, R. H.; et al. 2018. Data-driven Summarization of Scientific Articles. In *Proc. of the 7th International Workshop on Mining Scientific Publications, LREC 2018*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Passonneau, R. J.; and Litman, D. J. 1993. Intention-based segmentation: Human reliability and correlation with linguistic cues. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, 148–155. Association for Computational Linguistics.

Radev, D. R.; Jing, H.; Styś, M.; and Tam, D. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management* 40(6): 919–938.

Rambow, O.; Shrestha, L.; Chen, J.; and Lauridsen, C. 2004. Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*, 105–108.

Rossiello, G.; Basile, P.; Semeraro, G.; et al. 2017. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, 12–21.

Schuster, M.; and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45(11): 2673–2681.

See, A.; Liu, P. J.; Manning, C. D.; et al. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1073–1083.

Shang, G.; Ding, W.; Zhang, Z.; Tixier, A.; Meladianos, P.; Vazirgiannis, M.; and Lorré, J.-P. 2018. Unsupervised Abstractive Meeting Summarization with Multi-Sentence Compression and Budgeted Submodular Maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 664–674.

Sood, A.; Mohamed, T. P.; Varma, V.; et al. 2013. Topic-focused summarization of chat conversations. In *European Conference on Information Retrieval*, 800–803. Springer.

Sood, A.; Thanvir, M.; Vasudeva, V.; et al. 2012. Summarizing Online Conversations: A Machine Learning Approach. In *24th International Conference on Computational Linguistics-(Coling-2012)*.

Uthus, D. C.; and Aha, D. W. 2011. Plans toward automated chat summarization. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, 1–7.

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*, 5998–6008.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103.

Xie, S.; Liu, Y.; Lin, H.; et al. 2008. Evaluating the effectiveness of features and sampling in extractive meeting summarization. In *2008 IEEE Spoken Language Technology Workshop*, 157–160. IEEE.

Zechner, K. 2001. Automatic generation of concise summaries of spoken dialogues in unrestricted domains. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 199–207.

Zheng, H.; and Lapata, M. 2019. Sentence Centrality Revisited for Unsupervised Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6236–6247.

Zhou, L.; and Hovy, E. 2005. Digesting virtual "geek" culture: The summarization of technical internet relay chats. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 298–305.