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

¹School of Computer Science, Fudan University, Shanghai, China  
²Alibaba Group, China  
³Zhejiang University, Hangzhou, China  
⁴Indiana University Bloomington, Bloomington, United States

{yczou18, qz, xjhuang}@fudan.edu.cn, {linjun, lujun, 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).

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
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 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.

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.

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.

---

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

2https://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 \( D = \{u_1, u_2, \ldots, u_n\} \). To incorporate multi-party information, for the \( i \)-th utterance in a chat log, we have \( u_i = \{p_i, w_{i1}, \ldots, w_{im}\} \), 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:

\[
    h_i = \text{BERT}(u_i).
\]

**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 \( S_i = \{u_{i-c}, \ldots, u_i, \ldots, u_{i+c}\} \). 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:

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

where \( \mathcal{P}(u, u_j) = \sigma(h^T_{u_{ij}} W h_{u_i}) \) represents the utterance co-occurrence probability in a specific context scope, which can measure the relevance of two utterances. \( \mathcal{P}(u) \) is a uniform distribution from which we sample \( m \) negative examples for each positive data point. \( W \) is a trainable parameter. 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 sub-tokens. 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:

- 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 \( p_s \).
- With probability \( p_s \), 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 \( \hat{S}_i = \{\hat{u}_{i-c}, \ldots, \hat{u}_i, \ldots, \hat{u}_{i+c}\} \). All utterances in \( \hat{S}_i \) are also encoded by BERT as in Eq.1 where the BERT parameter weights are shared. The output representations are \( \hat{h}_i = [\hat{h}_{i-c}, \ldots, \hat{h}_i, \ldots, \hat{h}_{i+c}] \). 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 \( \hat{S}_i \). Here, we use a Transformer Encoder (Vaswani et al. 2017) to capture chat semantics and form queries \( q_i \) as follows:

\[
    [q_1, q_2, \ldots, q_n] = \text{TransEnc}([h_1, h_2, \ldots, h_n]).
\]

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

\[
    p(\hat{S}_i) = \text{TransDec}(\hat{H}_i; q_i).
\]

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

$$\mathcal{L}_{\text{rec}} = - \sum_i \log p(\hat{S}_i).$$

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

$$\mathcal{L} = \alpha \mathcal{L}_{\text{ce}} + (1 - \alpha) \mathcal{L}_{\text{rec}}.$$

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:

$$\tilde{M}_{ij} = \sigma(h_j^T W h_i).$$

Each score $\tilde{M}_{ij}$ 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 $\lambda^L$ to constrain the score of distant utterance pairs. According to the Gaussian distribution, we can get $\lambda^L$ and the final relevance matrix $M$ as follows:

$$\lambda^L_{ij} = \exp\left[-\frac{1}{2n^2} \sum_{k=1}^{n} \left(P_{jk} - P_{ik}\right)^2\right],$$

$$M_{ij} = \lambda^L_{ij} \tilde{M}_{ij},$$

where $1 \leq P_{ij}, P_{j} \leq n$ 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):

$$C(u_i) = \sum_{1 \leq j \leq n, j \neq i} M_{ij},$$

which can be called local centrality score since $\lambda^L$ 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:

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

where $\eta$ 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 $k$:

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

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 $x \in \mathcal{X}$, we input utterance representations $H$, namely $H = [h_{i-1}, ..., h_i, ..., h_{i+c}]$, and the query embedding $q_i$ into the Transformer decoder as follows:

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

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. **SummAEE** (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 about 3.12% of utterances were truncated.

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

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.

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.

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 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
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.

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

Table 2: Human evaluation results in relevance and succinctness. The score represents the percentage of times each method is chosen as better in pairwise comparisons.

<table>
<thead>
<tr>
<th>Models</th>
<th>Relevance</th>
<th>Succinctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextRank</td>
<td>37.8%</td>
<td>41.7%</td>
</tr>
<tr>
<td>SummAE</td>
<td>28.2%</td>
<td>45.7%</td>
</tr>
<tr>
<td>RankAE (Ext.)</td>
<td>39.3%</td>
<td>50.4%</td>
</tr>
<tr>
<td>RankAE</td>
<td>57.2%</td>
<td>46.4%</td>
</tr>
<tr>
<td>Gold</td>
<td>87.5%</td>
<td>65.8%</td>
</tr>
</tbody>
</table>

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 3 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 cover-
The price is 180 excluding tax and shipping. It 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 (Radav 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 GNUe IRCs or query terms from WordNet synonyms. Moreover, chats in the GNUe 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.

---

### Table 4: An example of chat summarization with RankAE.

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

Texts with red color represent nonessential or redundant content in the chat segment, which are excluded by RankAE to produce a more concise summary.

### 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 contents 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
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


