Building Multi-turn Query Interpreters for E-commercial Chatbots with Sparse-to-dense Attentive Modeling

Yan Fan, Chengyu Wang, Peng He, Yunhua Hu
Alibaba Group
Hangzhou, China
{fanyan.fy,chengyu.wcy,hepeng.hp,wugou.hyh}@alibaba-inc.com

ABSTRACT

Predicting query intents is crucial for understanding user demands in chatbots. In real-world applications, accurate query intent classification can be highly challenging as human-machine interactions are often conducted in multiple turns, which requires the models to capture related information from the entire contexts. In addition, query intents tend to be fine-grained (up to hundreds of classes), containing lots of casual chats without clear intents. Hence, it is difficult for standard transformer-based models to capture complicated language characteristics of dialogues to support these applications. In this demo, we present AliMeTerp, a multi-turn query interpretation system, which can be seamlessly integrated into e-commercial chatbots in order to generate appropriate responses. Specifically, in AliMeTerp, we introduce SAM-BERT, a pre-trained language model for fine-grained query intent understanding, based on Sparse-to-dense Attentive Modeling. For model pre-training, a stack of Sparse-to-dense Attentive Encoders are employed to model the complicated dialogue structures from different levels. We further design Hierarchical Multi-grained Classification tasks for model fine-tuning. Experiments show SAM-BERT consistently outperforms strong baselines over multiple multi-turn chatbot datasets. We further show how AliMeTerp is deployed in real-world e-commercial chatbots to support real-time customer service.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; *Learning latent representations.*

KEYWORDS

query intent understanding; multi-turn query interpretation; pre-trained language model; e-commercial chatbot

ACM Reference Format:
Yan Fan, Chengyu Wang, Peng He, Yunhua Hu. 2022. Building Multi-turn Query Interpreters for E-commercial Chatbots with Sparse-to-dense Attentive Modeling. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3488560.3502189

WSDM '22, February 21–25, 2022, Tempe, AZ, USA.

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3502189

Dialogue	*Query 1:* How much is the shipping fee? *Answer 1:* Free shipping over 39. *Query 2:* I meant the return of the product.
Incorrect Intent	Return & Refund
Response	*You can send the product back as long as the original packaging is undamaged.*
Correct Intent	Shipping Fee for Returns
Response	*You need to pay for the shipping fee if you return the product for personal reasons.*

Table 1: An example of multi-turn QIU. The texts are in Chinese and have been translated into English. Responses w.r.t. the predicted intents are printed in italics.

1 INTRODUCTION

Chatbots are ubiquitous systems that conduct conversations with humans, especially in e-commercial services [7]. In chatbots, Query Intent Understanding (QIU) is crucial for understanding needs of humans, in order to generate appropriate responses [9].

In the literature, extensive research has focused on dialogue state tracking in closed domains [10]. Different from previous studies, e-commercial chatbots in industry work in boarder domains, often associated with a fine-grained categorization of query intents [2]. For example, the number of query intents in the e-commercial chatbot of Alibaba Group called Alime[1] is over three hundred, including a special label `UNK` (meaning no clear intents expressed). Such queries are highly informal and fragmented, lacking standard grammatical structures for accurate semantic analysis. Additionally, interactions between humans and machines are often in multiple turns, with complicated dialogue structures involved. Consider the example in Table 1. The actual intent of the customer is expressed across multiple turns, rather than by a concrete sentence. Hence, it is necessary for chatbots to understand the meanings of the entire dialogue to predict the correct query intent. Recently, pre-trained language models such as BERT [3] have achieved state-of-the-art performance on various NLP tasks. However, these models are not optimized to capture contextual information from utterances in chatbots. Chao et al. [1] learn contextual representations of dialogues, but mostly focus on closed-domain, task-oriented scenarios.

In this demo, we present AliMeTerp, a multi-turn query interpreter that can be seamlessly integrated into e-commercial chatbots. AliMeTerp predicts query intents of the customer by analyzing the entire dialogue and then generates suitable responses by retrieving answers from the Knowledge Bases (KBs) of the underlying

[1] https://www.alixiaomi.com/

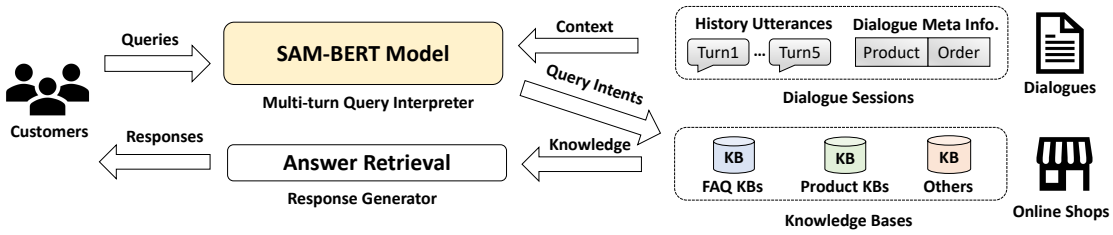

Figure 1: The general workflow of the AliMeTerp system.

online shops. In AliMeTerp, we design a BERT-style pre-trained language model for multi-turn QIU named SAM-BERT based on Sparse-to-dense Attentive Modeling. Here, we design the stacked Sparse-to-dense Attentive Encoders (SAEs) to model the dialogue characteristics from three levels, i.e., the utterance-level, the utterance pair-level and the entire dialogue-level. A fine-grained QIU model is obtained by fine-tuning SAM-BERT, which particularly handles queries without any intents and fine-grained semantics of queries. Our AliMeTerp system integrates the complete process of training, evaluating and deploying SAM-BERT models, together with the answer retrieval functionalities for e-commercial chatbots.

To verify the effectiveness of our approach, we evaluate SAM-BERT over multiple real-world multi-turn chatbot datasets. The results show that SAM-BERT consistently outperforms strong baselines. Currently, the AliMeTerp system is deployed in various business scenarios of Alibaba Group to provide better online shopping service through our AliMe chatbot. We will further demonstrate how AliMeTerp can benefit a variety of e-commercial chatbots and how SAM-BERT can be easily trained by our high-level APIs.

2 SYSTEM DESIGN

In this section, we first describe the system flow of AliMeTerp briefly. After that we introduce the *multi-turn*, *fine-grained* QIU task and the technical details of the SAM-BERT model.

2.1 Workflow of AliMeTerp

The system workflow of AliMeTerp is presented in Figure 1. When a customer issues a query to the chatbot, our system restores the entire dialogue session and employs SAM-BERT to predict the query intent. The meta information is also included in the session, such as the product or the order that the customer queries. For online shops, the owners may upload various types of KBs to the system, such as the FAQ KB and the product KB. The response generator returns the answer from certain KB with the corresponding intent to the customer (which is not our major focus). As seen, SAM-BERT is the most important part in AliMeTerp that controls the flows of the dialogue. In the following, we introduce the details of SAM-BERT.

2.2 Model Architecture of SAM-BERT

We start with some basic notations. Let $\mathcal{L}$ be a finite, pre-defined set of query intent labels, which includes a special label `UNK`, meaning no specific intent is associated with the corresponding query. To facilitate fine-grained QIU, we assume the size of $\mathcal{L}$ is large. Denote $\mathcal{S} = \{q_1, a_1, \cdots, q_k\}$ $(k > 0)$ as a dialogue session, where $q_1, \cdots, q_k$ and $a_1, \cdots, a_{k-1}$ are user and chatbot utterances, respectively. Let $D = \{(\mathcal{S}, l)\}$ be the training set, containing the dialogue $\mathcal{S}$ and the intent label $l \in \mathcal{L}$ pairs. The task is to train a classifier f from D that is capable of predicting the query intent label of the k-th query q_k in the session $\mathcal{S}$, given $q_1, a_1, \cdots, q_{k-1}, a_{k-1}$ as the context.

In SAM-BERT, as a pre-processing step, each dialogue session $\mathcal{S}$ is converted into a sequence of tokens: `[CLS][QUE]`$t(q_1)$`[ANS]` $t(a_1) \cdots$ `[QUS]`$t(q_k)$ where `[CLS]`, `[QUE]` and `[ANS]` are special tokens for the classification output, the starting positions of queries and answers, respectively. $t(q_n)$ and $t(a_n)$ are token sequences for the n-th query and answer in $\mathcal{S}$, respectively. In the input layer, the sequence of tokens are mapped to WordPiece, segment and positional embeddings [3]. After that, the embedding sequences are passed through K stacked Sparse-to-dense Attentive Encoders (SAEs) before it reaches to the final classifier f to generate the predicted query intent label $l \in \mathcal{L}$. Let $\vec{h}_m^{(p)}$ be the m-th token embeddings of the entire sequence generated by the p-th layer of SAEs. We have: $\vec{h}_m^p = \text{SAE}(\vec{h}_1^{p-1}, \cdots, \vec{h}_M^{p-1})$ where M is the maximum length of the sequence. Finally, we have the last-layer embeddings of all the tokens $\vec{h}_1^K, \cdots, \vec{h}_M^K$, which are used as features for prediction.

Sparse-to-dense Attentive Encoder. Let d be the dimension of output of the embedding layer. In each layer of SAEs, we randomly divide $\vec{h}_m^p$ into three parts, each with $\frac{d}{3}$ channels, which correspond to input embeddings of modeling from different levels of dialogue structures, denoted as $\vec{h}_{m,\alpha}^p$ (the utterance-level), $\vec{h}_{m,\beta}^p$ (the utterance pair-level) and $\vec{h}_{m,\gamma}^p$ (the dialogue-level).

For the utterance-level, the embeddings $\vec{h}_{m,\alpha}^p$ are the self-attentive output of $\vec{h}_{m-r,\alpha}^{p-1}, \cdots, \vec{h}_{m+s,\alpha}^{p-1}$, where tokens within the range $[m-r, m+s]$ form the same utterance (either the query or the answer with maximum length of $r+s$). The computation for the utterance pair-level $\vec{h}_{m,\beta}^p$ takes a little step forward w.r.t. the structure scope. The embeddings are the attentive summarization of all token embeddings from the previous SAE block within the scope of the same utterance pair. [2] The dialogue-level token embeddings $\vec{h}_{m,\gamma}^p$ are generated with the attention scope of the entire sequence. By concatenating the three types of token embeddings, we produce the m-th token emebddings of the p-th SAE layer: $\vec{h}_m^p = \vec{h}_{m,\alpha}^p \oplus \vec{h}_{m,\beta}^p \oplus \vec{h}_{m,\gamma}^p$. For the representations of the `[CLS]` token, we utilize full attention in all three cases. Refer to Figure 2 for the attention building blocks.

Pre-training SAM-BERT. Because SAM-BERT has different attentive structures from BERT [3], we do not use BERT checkpoints

[2] Note that token embeddings of the last query q_k in the utterance pair-level are computed by the same way as the utterance-level, because the answer to q_k is unavailable.

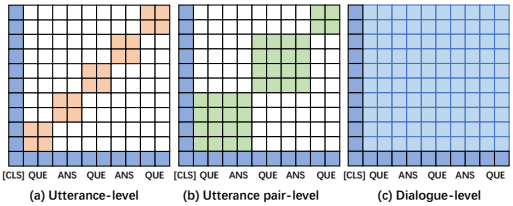

Figure 2: Building blocks of the sparse-to-dense attentions.

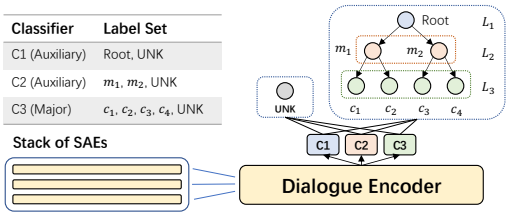

Figure 3: Fine-tuning SAM-BERT for fine-grained QIU.

to initialize SAM-BERT. Instead, we pre-train SAM-BERT over massive dialogue corpora from scratch by self-supervised learning.

For the pre-training task *masked language modeling* (MLM), we keep it the same as in Devlin et al. [3]. To better capture dialogue structures, the second task is *dialogue structure prediction* (DSP), which is a classification task. Three types of labels are introduced: i) *Origin*: the dialogue session is unchanged; ii) *Permutation*: we randomly inter-change two queries in the session; and iii) *Replacement*: We replace the last query (q_k) with one query from other sessions. The latter two types are particularly designed to learn the *semantic consistency* of the dialogue session and the *topical coherence* of utterance pairs. Overall, the pre-training loss is: $\mathcal{J}_{Pre} = \mathcal{J}_{MLM} + \lambda_1 \mathcal{J}_{DSP}$, where $\mathcal{J}_{MLM}$ is the MLM loss, $\mathcal{J}_{DSP}$ is the cross-entropy loss for DSP, and λ_1 is the balancing factor.

Fine-tuning SAM-BERT. To identify the large number of UNK labels and also address fine-grained QIU, the Hierarchical Multi-grained Classification (HMC) technique is proposed for fine-tuning. As different fine-grained intents have internal semantic associations, it is natural to group these intents into clusters. For each intent label $l \in \mathcal{L} \setminus \{\text{UNK}\}$, we represent l by the centroid embedding of its training instances, generated from the pre-trained SAM-BERT model. By applying hierarchical clustering, the tree-based semantic structure of query intents can be generated. Denote $\{\mathcal{L}_1, \cdots, \mathcal{L}_Q\}$ as the collections of nodes in the tree where $\mathcal{L}_i$ is node collection of the i-th layer, and Q is the total number of layers in the tree. Here, $\mathcal{L}_1$ only contains the root node and $\mathcal{L}_Q = \mathcal{L} \setminus \{\text{UNK}\}$. Refer to Figure 3 (with $Q = 3$). Denote $\mathcal{J}_{CLS}(\mathcal{S}, l)$ as the sample-wise cross-entropy loss of the final query intent classifier C_Q (with $|\mathcal{L}|$ classes). $\mathcal{J}^{(i)}_{CLS}(\mathcal{S}, l)$ is the auxiliary sample-wise cross-entropy loss for classifier C_i in label set $\mathcal{L}_i \cup \{\text{UNK}\}$. Apart from optimizing $\mathcal{J}_{CLS}(\mathcal{S}, l)$ directly, we add $Q-1$ auxiliary losses $\mathcal{J}^{(i)}_{CLS}(\mathcal{S}, l)$ ($i = 1, \cdots, Q-1$) to make the model to learn the semantic relations of fine-grained intents and the UNK labels. The total loss function $\mathcal{J}$ is defined as follows:

$$\mathcal{J} = \sum_{(\mathcal{S},l)\in\mathcal{D}} \mathcal{J}_{CLS}(\mathcal{S}, l) + \lambda_2 \sum_{(\mathcal{S},l)\in\mathcal{D}} \sum_{i=1}^{Q-1} \mathcal{J}^{(i)}_{CLS}(\mathcal{S}, l)$$

Dataset	# Train	# Dev	# Test	# Intents
JDDC	636,596	9,704	9,877	277
ChatGeneral	123,737	4,428	4,167	309
ChatFashion	46,737	2,586	2,523	307
ChatCosmetic	24,532	1,444	1,390	250

Table 2: Statistical summary of the four datasets.

where $\lambda_2 > 0$ is a tuned hyper-parameter.

3 SYSTEM EVALUATION

To verify the effectiveness of our system, we conduct extensive experiments to evaluate SAM-BERT over multiple datasets. After that, we present the A/B test of our method for online deployment.

3.1 Datasets and Experimental Settings

We conduct experiments on four large-scale datasets generated from chatbots. The first is a subset of the *JDDC Corpus* [2], which is a large-scale real-scenario dialogue corpus. We also construct three datasets from our in-house e-commercial chatbots in three domains, namely *ChatGeneral* (CG), *ChatFashion* (CF) and *ChatCosmetic* (CC), with statistics in Table 2. The ratios of the UNK labels of the four datasets are 0.45, 0.25, 0.32 and 0.40, respectively. We collect 3 million dialogues from our chatbots and 1 million dialogues from the JDDC corpus to pre-train SAM-BERT (together with other baseline language models), which contain over 25 million utterances. During pre-training, 15% of the words are masked for prediction. Utterances with *Origin*, *Permutation* and *Replacement* labels are sampled with the uniform distribution. The default parameter settings of SAM-BERT is as follows: $\lambda_1 = 0.9$, $\lambda_2 = 0.7$ and $Q = 3$. The hidden and embedding sizes are 756 and 128. We also tune the parameters over the development sets. With parameter sharing applied [5], the total number of parameters is 24M. We train the model with the Adam optimizer and the learning rate is 1e-5. All the algorithms are implemented in TensorFlow and run with 8 Tesla V100 GPUs.

3.2 Experimental Results

General Performance Comparison. For fair comparison, we consider TextCNN [4] and two-tower RNN [6] as non-PLM baselines. Here, our TextCNN implementation takes TextCNN to obtain representations of all utterances, and employs averaged pooling of representations to make predictions. Our two-tower RNN implementation encodes the context and the target query separately, and trains the classifier over both features. Several popular base-version PLMs are employed as strong baselines, namely ALBERT [5], BERT [3] and ToD-BERT [8], which is the state-of-the-art PLM for modeling dialogues. The general performance on four testing sets is in Table 3. SAM-BERT consistently performs better than other baselines, compared to the best competitor. Additionally, SAM-BERT has only 20% of the parameters compared to original BERT (24M vs. 110M). Hence, SAM-BERT is both more accurate in prediction and smaller in model size.

Detailed Model Analysis. We conduct ablation experiments to analyze various aspects of SAM-BERT. As for the model structure, we remove one type of attention at each time (i.e., $\vec{h}^p_{m,\alpha}$, $\vec{h}^p_{m,\beta}$ and $\vec{h}^p_{m,\gamma}$), and report the performance based on other two types. From Table 4, we see that utterance-pair level token embeddings ($\vec{h}^p_{m,\alpha}$)

Model	JDDC	ChatGeneral	ChatFashion	ChatCosmetic
TextCNN	0.6308	0.6179	0.6262	0.5728
RNN	0.6327	0.6136	0.6425	0.5847
ALBERT	0.7242	0.6939	0.7169	0.6546
BERT	0.7380	0.7448	0.7490	0.6899
ToD-BERT	0.7342	0.7455	0.7595	0.6939
SAM-BERT	**0.7524**	**0.7629**	**0.7715**	**0.7057**

Table 3: Testing performance of SAM-BERT and baseline models over four datasets, in terms of F1-score.

Ablation	Model Variants	F1-score
Model Structure	w/o. utterance level attention	0.7461
	w/o. utterance-pair level attention	0.7428
	w/o. dialogue level attention	0.7462
Feature Space	[CLS] token	0.7067
	[CLS] token + token embeddings	0.7583
	[CLS] token + CNN sub-network	**0.7629**

Table 4: Ablation study on the *ChatGeneral* dataset.

System	Accuracy	Improvement
Online System	0.72	-
SAM-BERT	**0.85**	**+13%**

Table 5: The online deployment performance (A/B test) of SAM-BERT in terms of accuracy.

contribute the most. We further consider which set of features is more useful for prediction. Four settings are considered: i) the [CLS] embeddings only (i.e., $\vec{h}_1^K$); ii) the [CLS] embeddings and the averaged all token embeddings; and iii) the [CLS] embeddings, together with a CNN-based sub-networks with all embeddings as features. The results show that combining a simple CNN model of all embeddings has the best results.

Online Deployment. The AliMeTerp system has been deployed online to produce intelligent customer service in Alibaba Group. Note that the existing online production system is a TextCNN model, which is able to respond within 3ms for 95% of the requests under the demand of high performance of the online system. Here, we report the online A/B test results of SAM-BERT, show in Table 5. From the results, we can see that SAM-BERT significantly improves the accuracy by 13%, which is a large margin. Currently, we have deployed our framework to provide service for over 48,000 online shops, benefiting millions of customers.

4 DEMONSTRATION SCENARIOS

In this demonstration, we will showcase the entire business process of AliMeTerp. Some snapshots of AliMeTerp can be found in Figure 4. As seen, the customers can conduct multi-turn conversations with our chatbots to fulfill their online shopping experience. Additionally, in AliMeTerp, the staff of online shops can manage their KBs and guide the chatbots to generate suitable responses based on the predicted query intents by SAM-BERT.

Because SAM-BERT is highly useful to model the dialogue semantics, for both pre-training language models with massive online chat logs and also fine-tuning for fine-grained intents in real-world chatbot scenarios. We will release the toolkit to public, namely sambertcmd. With only one line of command, developers can train

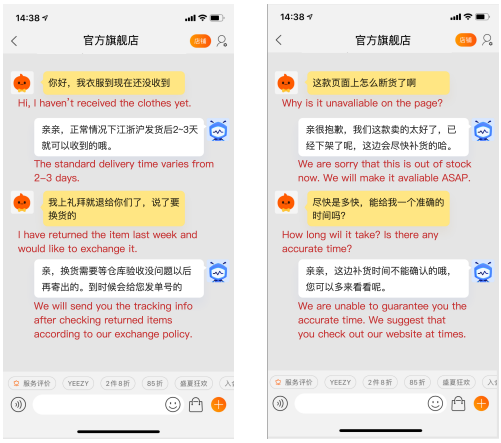

Figure 4: The snapshots of AliMeTerp used in AliMe chatbot.

Training & Evaluation

```
sambertcmd --mode train \
  --input=train.csv,dev.csv\
  --inputSchema=context:str:1,query:str:1,label:str:1 \
  --contextSequence=context --querySequence=query \
  --sequenceLength=128 \
  --labelName=label \
  --checkpointDir=./trained_models/ \
  --numEpochs=3 --batchSize=32 \
  --optimizerType=adam --learningRate=1e-5
```

Model Prediction

```
sambertcmd --mode predict \
  --input=test.csv --output=test.pred.csv \
  --inputSchema=id:str:1,context:str:1,query:str:1 \
  --contextSequence=context --querySequence=query \
  --outputSchema=predictions,probabilities,logits \
  --checkpointPath=./trained_models/
```

Figure 5: The command-line APIs for running sambertcmd.

and evaluate the SAM-BERT model on GPU servers, together with model inference. Readers can refer to Figure 5 for APIs.

REFERENCES

[1] Guan-Lin Chao and Ian Lane. 2019. BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer. In *Interspeech*. 1468–1472.

[2] Meng Chen, Ruixue Liu, Lei Shen, Shaozu Yuan, Jingyan Zhou, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. The JDDC Corpus: A Large-Scale Multi-Turn Chinese Dialogue Dataset for E-commerce Customer Service. In *LREC*. 459–466.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.

[4] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.

[5] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR*.

[6] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI*. 2786–2792.

[7] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. In *ACL*. 498–503.

[8] Chien-Sheng Wu, Steven C. H. Hoi, Richard Socher, and Caiming Xiong. 2020. ToD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogues. *CoRR* abs/2004.06871 (2020).

[9] Liu Yang, Minghui Qiu, Chen Qu, Cen Chen, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, and Haiqing Chen. 2020. IART: Intent-aware Response Ranking with Transformers in Information-seeking Conversation Systems. In *WWW*. 2592–2598.

[10] Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-Locally Self-Attentive Encoder for Dialogue State Tracking. In *ACL*. 1458–1467.