Question-Answer Pairing from IM Conversations via Message Merging and Reply-to Prediction

Thamolwan Poopradubsil
Department of Computer Science
Kasetsart University, Thailand
tmw.poopradubsil@gmail.com

Chia-Hui Chang
Department of Computer Science
and Information Engineering
National Central University, Taiwan
chia@csie.ncu.edu.tw

Abstract

Preparing question-answer pairs from conversation logs (chat logs) is often considered a prerequisite for downstream dialogue tasks such as response generation and response selection tasks. In this paper, we study a task called reply-to prediction, which can be used to prepare question-answer pairs. Unlike other works, our data comes from the instant messaging (IM) platform where participants could split long sentences into short utterances and send them in multiple messages. We consider a task called message merging task which aims to determine whether those messages need to be merged or not before generating message pairs for reply-to prediction task. The theory behind this task is similar to, yet different from reply-to prediction task in which this task uses the messages from the same speaker to predict whether these two messages are related or not. We propose a CONTEXT-AOA model to include the context (previous dialogue) as additional input apart from pairwise messages. Our experiments show that our proposed model outperforms both single-turn (pairwise) conversation models and multi-turn (context-aware) conversation models on message merging task and achieves a close performance compares to other multi-turn models on reply-to prediction for manually labeled data and outperforms other models when using heuristic labeled data.

1 Introduction

Preparation of questions-answers from conversation logs (also known as chat logs) between customers and enterprise personnel is important for the development of chatbots. For example, Figure 1 shows a conversation between a customer service staff and a client. The goal here is to find the question-answer pairs such as (d, e), (p, q), from the chat logs such that the question-answer pairs can be used as the chatbot's responses to the corresponding question.

Finding the question-answer pairs is similar to the link prediction task where the goal is to decide whether an incoming message replies to the existing question based on the similarity between messages. Link prediction can be defined either as a reply-to prediction or as a response selection. Both of the problems require message pair similarities to determine which previous or following message is the best-matched candidate question or response. Similar to conversation disentanglement, earlier works on response selection only use the last message in the context for matching with the candidate response message from different utterances (Wang et al., 2013). However, human does not give the response entirely based on a link prediction task. Real-world chat logs are multi-turn conversations, thus including the context as another input is necessary in order to allow a model to remember what has been discussed so far. Recent works show that including the multi-turn conversation improves the response selection's performance, e.g. (Zhou et al., 2016; Zhou et al., 2018; Wu et al., 2019).

In this work, we focus on online instant messages between customers and service personnel. The major problem is that a complete message (either the question or the answer) may be split into several ut-

Data Set			Manual		Heuristic	
ID	Author	Messages	ID_M	RT	ID_H	RT
a	Client	need support with an open ticket	0		0	
b	QNAP	paoyang	1		1	0
c	QNAP	Hi, how are you doing				
d	QNAP	what is the ticket #?	2	0		
e	Client	#FNA-202-51465	3	2	2	1
f	Client	it is for 3M				
g	Client	they are very unhappy with the ...				
h	Client	ive been on hold on the phone ...				
i	QNAP	I am working on that ticket.	4	3	3	2
j	QNAP	They are using an embedded ...				
k	QNAP	According to the tech who worked on it ...				
l	Client	have you let them know ...	5	4	4	3
n	Client	so will it not work at all with their server?				
m	QNAP	so this seems like an issue with ...	6		5	4
o	QNAP	It seems to be closely related to ...				
p	Client	So when should I tell ...	7	6	6	5
q	QNAP	I will respond back to them but ...	8	7	7	6
r	Client	Okay, I will let them know of our conversation.	9	8	8	7
s	Client	Thank you.				

Figure 1: A real-world conversation between QNAP customer service and client.

terances. For example, e, f, g and h in Figure 1 together make a complete answer to question d, therefore these four messages should be merged together into one utterance. However, we cannot directly merge all adjacent utterances from the same speaker since each message might be either an incomplete message related to its previous utterance or a complete message on its own. For example, c and d should not be merged since message d alone makes a complete response to message a while message c asks about something unrelated to the question in message a.

To deal with this kind of dataset, we propose a two-stage pipeline for the question-answer pair preparation. We distinguish the task of extracting question-answer pairs into two subtasks: message merging task and reply-to prediction task in order to solve the problem we mentioned previously. We first regard both subtasks as a sentence pair classification task within a single-turn conversation setting. Three neural networks models based on GloVe word embedding (including CNN+LSTM, LSTM with dual attention, and attention over attention (AOA) (Huang et al., 2018)) as well as two BERT (Devlin et al., 2019) based models (BERT sentence pair classification and the combination of BERT-SPC and AOA) are proposed. However, the best performance only achieves 0.763 and 0.794 accuracy (0.443 and 0.498 F1) on the message merging task and reply-to prediction task respectively.

To further improve the performance, we consider including context (previous dialogue) as additional input apart from only pairwise messages. With the application of AOA over any pairs of the context and two messages, we show a significant improvement over the previous models for message merging (0.964 accuracy and 0.887 F1) and even outperform existing models on reply-to prediction task when using heuristic labeled data (0.956 accuracy and 0.823 F1).

The rest of the paper is organized as follows: First, we give a definition of the two tasks (message merging task and reply-to prediction task)r. Then, we introduce the dataset used in this paper and describe the preparation process. Next, we describe our proposed model (CONTEXT-AOA). Finally, we

show the question-answer pair's preparation model for both tasks and compare them with the results from existing models.

2 Related Work

2.1 Conversation Disentanglement

The idea of treating the disentanglement task as a two-stage problem which has later been the major idea for most existing studies on this task has been proposed by Elsner and Charniak (2008). The first stage is a binary classification task where the main goal is to mark each message pair as alike or different while the second stage utilizes the results from the first stage to cluster the utterances into each conversation thread (same thread prediction).

Earlier approaches to the first stage or link (reply-to) prediction are based on a statistical classifier with the use of linguistic features in order to calculate the similarity of each message pair, e.g. (Elsner and Charniak, 2008; Elsner and Charniak, 2010). Recent approaches use neural models to learn abstract linguistic features. Mehri and Carenini (2017) use handcraft features along with the probability of being the next utterance which is predicted from a pre-trained LSTM model to train the reply classifier. Jiang et al. (2018) introduced Siamese Hierarchical CNN (SHCNN) to investigate how message similarity could be estimated. SHCNN uses hierarchical CNN to capture both low-level and high-level message meanings of each message. The interaction between two message representations which is computed using element-wise absolute difference methods is then used along with the handcraft features to estimate the similarity between two messages.

2.2 Single-turn vs. Multi-turn Response Selection

Earlier works on response selection tasks are only based on pairwise message comparison. Lu and Li (2013) proposed a DNN-based matching model for short text response selection by extracting the local pairwise relations on a low level with patches and sending them to the DNN layers. Hu et al. (2014) utilize deep CNN in order to capture the interaction between message and response.

However, using only pairwise messages might not be enough to solve this task, Lowe et al. (2015) introduced the task called multi-turn conversation response selection pair requires matching between a response and a conversation context (previous utterances) instead of a single previous message. They applied LSTM on the concatenated utterances (context) and a response message to perform context-response matching on a word-level context vector. Later, the work by Zhou et al. (2016) extended this idea by performing context-response matching on noa t only the general word level context vector but also the utterance level context vector. However, ignoring the relationships between the utterances (context) when concatenating them together (Lowe et al., 2015) or converting the whole context to a vector without enough supervision from responses (Zhou et al., 2016) might be the cause of some information loss which affects the model performance. To address this problem, the work by Wu et al. (2017) proposed a sequential convolutional network (SCN) that matches a response with each utterance in the context with a CNN and then accumulated the matching vectors in the utterances' temporal order to model their relationships. Another work from Wu et al. (2019) replaced the convolution neural network (CNN) with an attention layer to extract the interaction between each utterance and response.

2.3 Message Pair Classification

To build a better model for reply prediction tasks, we also refer to other tasks that accept two messages as input such as aspect-based sentiment analysis (ABSA) and natural language inference (NLI).

Aspect-based sentiment analysis aims to determine the sentiment polarity of a given sentence and aspect. Many models and methods have been proposed from traditional machine learning methods (Kiritchenko et al., 2014) to deep learning models (Liu et al., 2020). For example, Wang et al. (2016) proposed an attention-based LSTM network for aspect-based sentiment classification. Huang et al. (2018) introduced an attention-over-attention (AOA) neural network to capture the interaction between aspects and context sentences, which outperform LSTM-based architectures. However, one drawback of LSTM-based architectures is that their training process is time-consuming. To address this problem, Xue and Li (2018) introduced GCAE

model where its main components are CNN and gating mechanisms to reduce the number of training parameters and speed up the procedure.

On the other hand, the task of natural language inference is to determine if one given statement (a premise) semantically entails another given statement (a hypothesis). For example, Parikh et al. (2016) proposed "Decomposable Attention Model" which uses a shared sentence representation with fewer parameters and mutual attention mechanism to build a model with high performance.

3 Problem Definition and Dataset

Instant messaging (IM) is a type of online chat which offers real-time text-based communication in which two persons participate in a single conversation over their computers or mobile devices within an Internet-based chatroom. This type of online chat is commonly used in many business companies in order to support their clients. Companies often store their conversation logs between customer service staff and clients in order to be used in the future to improve their service and customer experiences.

The main goal of this work is to predict whether one message is a reply-to message to a previous message. However, unlike messages from other sources such as email or Reddit forums, messages from instant messaging are generally short to enable quick response. Moreover, participants could split long sentences into short utterances and send them in multiple messages. Therefore, some messages need to be merged before generating reply-to message pairs.

The overview of the training phase is outlined as follows, where we build a merging model to decide whether two messages from the same speaker need to be merged and a reply-to model to decide whether two messages from two speakers are a reply-to message pair.

- **Message merging task** aims to determine whether a given message pair from the same speaker should be merged or not. Most continuous messages from the same speaker could be merged, however, some messages should not be merged when they deliver different intentions. As shown in Figure 1, message e to h can to be merged in order to make a complete response to message d, while message d should not be merged with message b and c because they serve different purposes.

- **Reply-to prediction task** is to determine whether a given message pair from different speakers is a question-answer pair or not. For example, the message $ID_M = 2$ is a response to message $ID_M = 0$ (positive) while the merged message $ID_M = 1$ is not a response to message message $ID_M = 0$ (negative).

3.1 Training Data Preparation

An in-house QNAP customer service dialogue dataset contains conversation logs between two participants: a QNAP client and a customer service staff. QNAP customer service staff may reply to previous messages for troubleshooting or ask questions to clarify the problem while assisting the clients. The chatlog disentanglement task here is to find direct reply messages for question-answer pairing. A total of 1,860 conversations with an average of 31.7 messages per conversation are collected as our dataset.

Manual Annotation

We randomly select 60 conversations and ask three annotators to label these conversations. Two of them are asked to label the 60 conversations while the third annotator is asked to make the final decision on the inconsistent labels. For the message merging task, we only focus on the message pairs where both two messages are from the same speaker. The merged message is renumbered with message ID ID_M. For the reply-to prediction task, the annotators will focus on the merged messages and mark the current message as a response to which previous ID_M from the other speaker.

Message Pairs Preparation Given the annotated messages, we can pair messages that need to be merged as a positive example and independent messages as a negative example. Since each message (2,136 messages for the message merging task and 1,366 messages for the reply-to prediction task) can be paired with all of its previous messages in the same conversation, the number of negative examples is much larger than that of positive examples. Thus, the kappa value from the two annotators is quite high

(0.876 and 0.990) for the merging and reply-to tasks, respectively.

Dataset	Chat Log		
Conversations	60		1,800
Labeling	Manual	Heuristic	Heuristic
# Messages	2,136	2,136	56,792
Pos. Pairs	770	1,082	29,437
Neg. Pairs	3,679	-	-
# Pairs	4,449	-	-
# Merged Msg	1,366	1,054	27,355
Reply-to	743	753	19,188
Non-Reply	4,032	3,407	86,290
# Pairs	4,775	4,160	105,478

Table 1: Training and Testing data in chat log

With the third annotators, we get a total of 770 positive pairs and 743 reply-to pairs. To deal with imbalanced data, we restrict the maximum number of negative message pairs for each message. That is to say, we conduct negative message pairs down sampling. For the message merging task, we set the maximum number of negative message pairs equal to 2 which means we randomly choose at most 2 negative pairs from all possible negative pairs for each message, and for the reply-to prediction task, we set the maximum number of negative message pairs equal to 4. Overall, we have 3,679 negative pairs for the message merging task and 4,032 non-reply pairs for the reply-to prediction task from the 60 conversations via manual labeling.

Heuristic Labeling

The heuristic labeling data is prepared by merging all consecutive utterances from the same speaker into one message under the assumption that a client service staff's message following the customer's question in the previous turn is a reply to the question and using this assumption to annotate all conversations automatically. As shown in Figure 1, adjacent messages from the same speaker are merged into one message and are renumbered with a new ID, ID_H. Since the heuristic labeling always merges the adjacent messages from the same speaker, there are no negative examples. We then generate message pairs $x = (i, j)$ with $i < j$, where x is a positive reply-to example if j equals $i + 1$, otherwise a negative (Non-Reply) example. Similar to the manual labeling process, we conduct negative example down sampling to prepare pairs. Finally, we generate 19,188 reply-to pairs and 86,290 non-reply pairs from 1,800 conversations.

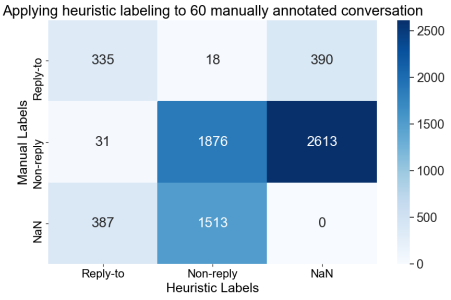

Figure 2: Applying heuristic labeling to 60 manually annotated conversations.

To see how effective the heuristic labeling is, we take the 60 manually annotated conversations as golden answers and compare them with the heuristic labeling result for performance evaluation. Since heuristic labeling merges all consecutive messages by the same speaker, the merge ratio (0.51=1082/2136) is higher than that of manual labeling (0.36=770/2136) (see Table 1). Due to the difference in the message merging step, some messages do not have corresponding matches in the other labeling method. Therefore, a third class "NaN" is used to denote message pairs that do not have corresponding matches as shown in Figure 2. Excluding unmatched pairs, the heuristic labeling has a 0.92 (=335/(335+31)) precision and 0.95 (=335/(335+18)) recall.

4 Context-Aware Message Pair Classification Models

Both the message merging and reply-to prediction models can be regarded as sentence pair classification models. However, the existing models which use only question-answer pair as input might not be enough for a task such as chat log (conversation) disentanglement. Inspired by the works on response selection task and AOA (Huang et al., 2018), we propose a model which includes the previous dialogue (context) as an input in addition to the question-answer pair. The overall model structure is as shown in Figure 3.

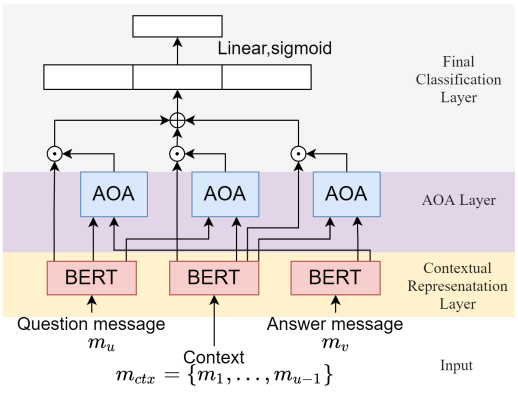

Figure 3: Context Attention-over-Attention BERT model structure ($u < v$).

Contextual Representation Layer

For each message pair (m_u,m_v) from a chat log C, we also include its context $m_{ctx} = \{m_1, ..., m_{u-1}\}$ as part of the input. That is to say, each training example is a triplet tuple $\mathbf{x} = (m_{ctx}, m_u, m_v)$. If m_v is a reply to m_u, $\mathbf{x}$ is considered a positive example, otherwise it is a negative example. We then apply shared BERT embedding to get the representation $h \in R^{l \times d}$ of each message m respectively, where l is the number of tokens after BERT word piece subword segmentation and d (=768) is the dimension size of BERT embedding.

Attention-over-Attention (AOA) Layer

Given two message representation $M_1 \in R^{n \times d}$ and $M_2 \in R^{m \times d}$, AOA first calculates a pair-wise interaction matrix $I = M_1 \cdot M_2^T$, where the value of each entry I_{ij} represents the correlation of a word pair among the two input messages. Next, two matrix column-wise softmax, $\alpha \in R^{n \times m}$ and row-wise softmax, $\beta \in R^{n \times m}$ are computed as follows.

$$\alpha_{ij} = \frac{exp(I_{ij})}{\sum_{k=1}^{n} exp(I_{kj})}, \beta_{ij} = \frac{exp(I_{ij})}{\sum_{k=1}^{m} exp(I_{ik})}, \tag{1}$$

The idea of AOA is to use the averaged attention weight $\overline{\beta} \in \mathbb{R}^m$ for the computation of output feature vector $\gamma \in \mathbb{R}^n$, where

$$\overline{\beta}_j = \frac{1}{n} \sum_{i=1}^{n} \beta_{ij}, \tag{2}$$

and the output of the attention-over-attention layer structure is computed by using $\overline{\beta}_j$ as a weight for each α_j:

$$AOA(M_1, M_2) = \alpha \cdot \overline{\beta}^{\mathsf{T}}. \tag{3}$$

Suppose the output of BERT embedding for the training example is denoted as M_u, M_v, and M_{ctx}. We then pair these embedding and apply attention-over-attention over three pairs to obtain $AOA(M_{ctx}, M_u)$, $AOA(M_{ctx}, M_v)$ and $AOA(M_u, M_v)$.

Final Classification Layer

Next, we use $AOA(M_1, M_2)$ for calculating the attention-weighted representations of each input pair.

$$r(M_u, M_v) = M_u^{\mathsf{T}} \cdot AOA(M_u, M_v)$$
$$r(M_{ctx}, M_u) = M_{ctx}^{\mathsf{T}} \cdot AOA(M_{ctx}, M_u)$$
$$r(M_{ctx}, M_v) = M_{ctx}^{\mathsf{T}} \cdot AOA(M_{ctx}, M_v)$$

Finally, we concatenate all the attention-weighted representations to the prediction layer, i.e. $\mathbf{p_o} = r(M_u, M_v) \oplus r(M_{ctx}, M_v) \oplus r(M_{ctx}, M_u)$ by Eq. 4.

$$P(y|x) = \sigma(\mathbf{w} \cdot \mathbf{p_o} + b_o) \tag{4}$$

5 Experiments

During testing time, we are given a chat log that is not labeled. We simply apply the merging model and reply-to prediction model in order as shown in Figure 4:

1. **Message pairing (Same speaker)**: We first pair the messages from the same speaker based on the trained merging models to determine whether these message pairs should be merged or not. The messages will be merged according to the output from the merging model. We then update the chat log file by replacing the message pairs that need to be merged with the merged messages.

2. **Message pairing (Different speaker)**: We then pair reply-to message pairs using the chat log we obtained in the previous step. Unlike the

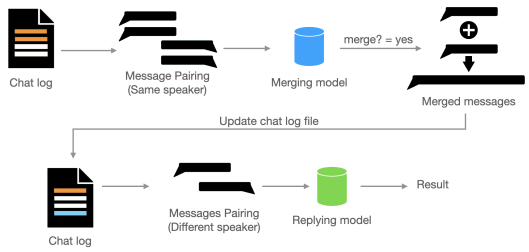

Figure 4: The testing phase of reply message prediction model.

first message pairing step, in this step, we focus on the message pairs from a different speaker. These pairs are then given to one of the reply-to prediction models to decide whether they are correct reply-to message pairs or not.

5.1 Experimental Setup

We divide the manually labeled data into 5-fold and use either 4-fold (48 conversations) out of the 60 conversations as training data to build the prediction models (both merging model and reply-to model for manually labeled examples). The models are tested on the remaining 12 conversations. The process is repeated five times and the result is averaged to obtain the final result. For heuristically merged and labeled examples from 1,800 conversations, we train the reply-to prediction models and test on all 60 conversations to compare the performance.

We implement three GloVe-based neural network models and two BERT models for performance comparison.

GloVe-based Models

A typical neural network model consists of an embedding layer for word representation, a hidden layer such as mutual attention for message representation, and an output layer for prediction. For the embedding layer, we adopt a pre-trained GloVe (Pennington et al., 2014) word embedding matrix from the Common Crawl dataset (42B tokens), which contains a case-sensitive vocabulary of size 1.9 million. We consider three models for message representation. The first one is GCNN-LSTM, the second is LSTM with dual attention, and the third is Attention-over-Attention (AOA) model.

- **GCNN-LSTM Representation** We use Convolutional Neural Networks (CNN) for feature extraction with Gated Linear Unit (GLU) proposed in (Dauphin et al., 2017) to control which information flows in the network. To deal with word sequence, we adopt a BiLSTM layer to capture the message information. The outputs from the BiLSTM layer are passed through two fully connected layers to make the prediction.

- **LSTM Dual Attention Model** Inspired by the power of the attention mechanism, the second model we proposed is BiLSTM with dual attention where we can exploit the attention mechanism to generate a representation for m_1 based on the content of m_2.

- **Attention-over-Attention (AOA) model** The above two models only focus on message representation. Therefore, we exploit the idea of capturing the interaction between one message to another message given the hidden semantic representations of the two messages generated by BiLSTMs with AOA. (Huang et al., 2018).

The pre-trained GloVe word embedding has a dimension size of 300. The hidden layers in BiLSTMs are 128, 128, and 300 for GCNN+LSTM model, LSTM+DualAtt model, and AOA model respectively, the number of kernels used in CNN is 128 with the kernel size equal to 5. The batch size used in the traditional deep learning model is 128 and the maximum epoch and initial learning rate are set to 40 and $1 * 10^{-3}$.

BERT-based Models

Different from context-free models, which generate a fixed word embedding representation for each word in the vocabulary, BERT is able to give a context-dependent representation of the words. Consequently, we use the BERT model released by Google and fine-tune it for the message merging/reply-to prediction task.

Given two input messages m_1 (with length n) and m_2 (with length m), we employ BERT component with L transformer layers to calculate the corresponding contextualized representations with input of the form $([CLS], m_1, [SEP], m_2)$.

- **BERT-SPC** The basic BERT sentence pair classification (BERT-SPC) model takes the output of [CLS] token as the prediction layer input.
- **BERT-SPC-AOA** We exploit the idea of Attention-Over-Attention model to further improve the BERT-SPC model by concatenating the output from AOA with [CLS] output as the input to the prediction layer.

For BERT based model, the batch size is 16 and 8 for CONTEXT-AOA model. The maximum epoch and initial learning rate are 6 and $2 * 10^{-5}$, respectively. The optimizer used in all models is Adam with $\beta 1 = 0.9$ and $\beta 2 = 0.999$. All models are trained on GeForce GTX1080Ti 10GB GPU.

5.2 Performance Comparison

Table 2 and 3 show the performance comparison of the proposed CONTEXT-AOA model with both single-turn models based on message pair similarity and multi-turn chatlog disentanglement models with additional context.

Message Merging Task

For message merging task, single-turn approaches with only two message input exhibit limited performance. The highest F1 score of these models is only 0.443 F1, which is achieved by BERT-SPC-AOA model as shown in Table 2, While all multi-turn approaches including (Lowe et al., 2015), (Zhou et al., 2016), and (Wu et al., 2019) have significant improvement over single turn approaches. The proposed CONTEXT-AOA model achieves the best 0.887 F1 and 0.964 accuracy.

	QNAP: Message Merging Task		
	Model	F1	Acc
Single-turn	SHCNN (Jiang et al., 2018)	0.266	0.763
	GCNN+LSTM	0.271	0.731
	LSTM+DualAtt	0.254	0.680
	AOA (Huang et al., 2018)	0.333	0.516
	BERT-SPC	0.374	0.734
	BERT-SPC-AOA	0.443	0.731
Multi-turn	LSTM (Lowe et al., 2015)	0.859	0.958
	MultiView (Zhou et al., 2016)	0.841	0.944
	SAN (Wu et al., 2019)	0.851	0.948
	CONTEXT-AOA	**0.887**	**0.964**

Table 2: QNAP chat log: Message Merging Task

Reply-to Prediction Task

For reply-to prediction task, we see a similar result. The proposed CONTEXT-AOA model yields 0.800 F1 and 0.944 accuracy, while the highest F1 score of single-turn models is 0.498 (by BERT-SPC-AOA model) as shown in the "Manual" column of Table 3. The experimental results demonstrate that one cannot neglect the relationship between previous messages (context) and the question-answer pair. Including previous messages as additional input significantly improves the performance for both subtasks on manually labeled examples.

For the result shown in the "Heuristic" column of Table 3, we train the models using all of the heuristically labeled data as training data and test on manually labeled examples where we divided the testing data into 5 folds and train the models similar to what we've done with the experiments on manually labeled examples.

Interestingly, the heuristically labeled data provides better performance for most of the multi-turn reply-to prediction models. The result may be attributed to a large amount of training data even though the heuristic labeling rule does not always generate the correct labeled data. This might also be the cause of an unstable performance for several models.

Figure 5 shows the performance of multi-turn conversation task models on heuristic labeled data in regard to the size of training data. We find that all models exhibit a steep slope the training data size is lower than 5%. Moreover, using 25%-50% of heuristic labeled data to train the models can significantly outperform the full size of manually labeled data, which is about 5% of the heuristic labeled data.

6 Conclusion

This paper addresses the problem of question-answer pairs preparation from two participants' online chat logs. The major problem with this kind of data is that a complete message may be split into several utterances, therefore additional task (message merging task) is required to merge some of these utterances together before forming the question-answer pairs. To extract question-answer pairs from chat logs, we perform reply-to prediction task on merged messages in order to identify the

	QNAP: Reply-to prediction				
	Data Set		Manual		Heuristic
	Model		F1	Acc	F1	Acc
Single-turn	SHCNN (Jiang et al., 2018)		0.429	0.722	0.432	0.702
	GCNN+LSTM		0.390	0.680	0.441	0.694
	LSTM+DualAtt		0.403	0.706	0.461	0.710
	AOA (Huang et al., 2018)		0.449	0.754	0.394	0.709
	BERT-SPC		shown1	0.736	0.413	0.665
	BERT-SPC-AOA		0.498	0.794	0.410	0.717
Multi-turn	LSTM (Lowe et al., 2015)		0.750	0.932	0.796	0.946
	MultiView (Zhou et al., 2016)		0.802	0.943	0.820	0.950
	SAN (Wu et al., 2019)		**0.814**	**0.948**	0.811	0.952
	CONTEXT-AOA		0.800	0.944	**0.823**	**0.956**

Table 3: QNAP chat log: Reply-to prediction

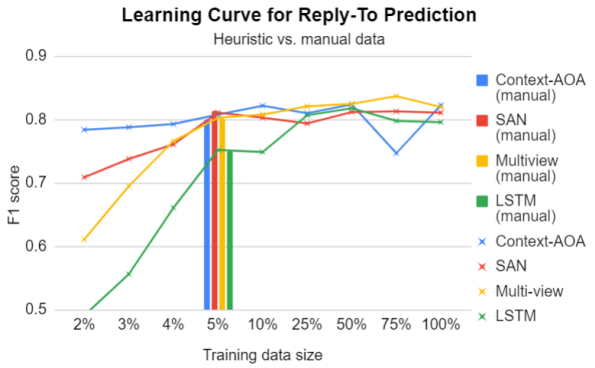

Figure 5: Learning curve for reply-to prediction task.

correct question-answer pairs. In terms of model design, we propose a context-aware AOA model which utilizes the idea of Attention-over-attention models to capture the relationship between context, question, and answer message.

Experimental results on both message merging and reply-to prediction tasks show that allowing the model to gain access to the context significantly improves the performance on both tasks. Our proposed CONTEXT-AOA model outperforms the existing models on message merging task and achieves comparable performance on reply-to prediction task for manually labeled data. In addition to manually labeled data, we also conduct experiments on heuristically labeled data where our proposed model outperforms the existing models and the result demonstrates that more training data may further improve the problem.

Acknowledgments

The research is partially supported by Ministry of Science and Technology, Taiwan under grant MOST109-2221-E-008-060-MY3.

References

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 933–941. JMLR.org.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Columbus, Ohio, June. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2010. Disentangling chat. *Computational Linguistics*, 36(3):389–409.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2042–2050. Curran Associates, Inc.

Binxuan Huang, Yanglan Ou, and Kathleen M. Carley. 2018. Aspect level sentiment classification with attention-over-attention neural networks. *CoRR*, abs/1804.06536.

Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. 2018. Learning to disentangle interleaved conversational threads with a Siamese hierarchical network and similarity ranking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1812–1822, New Orleans, Louisiana, June. Association for Computational Linguistics.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442, Dublin, Ireland, August. Association for Computational Linguistics.

H. Liu, I. Chatterjee, M. Zhou, X. S. Lu, and A. Abusorrah. 2020. Aspect-based sentiment analysis: A survey of deep learning methods. *IEEE Transactions on Computational Social Systems*, 7(6):1358–1375.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic, September. Association for Computational Linguistics.

Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 1367–1375. Curran Associates, Inc.

Shikib Mehri and Giuseppe Carenini. 2017. Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 615–623, Taipei, Taiwan, November. Asian Federation of Natural Language Processing.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, November. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 935–945, Seattle, Washington, USA, October. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas, November. Association for Computational Linguistics.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505, Vancouver, Canada, July. Association for Computational Linguistics.

Yu Wu, Wei Wu, Chen Xing, Can Xu, Zhoujun Li, and Ming Zhou. 2019. A sequential matching framework for multi-turn response selection in retrieval-based chatbots. *Computational Linguistics*, 45(1):163–197, March.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523, Melbourne, Australia, July. Association for Computational Linguistics.

Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 372–381, Austin, Texas, November. Association for Computational Linguistics.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127, Melbourne, Australia, July. Association for Computational Linguistics.