Chat-log Disentanglement via Same-Thread Classification and Direct-Reply Prediction

Chia-Hui Chang, Zhi-Xian Liu
Yu-Ching Liao, Yu-Hao Wu
National Central University, Taiwan
chia@csie.ncu.edu.tw

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

Abstract

The early purpose of chatlog (conversation) disentanglement is to separate intermingled messages into detached conversations for easier information following and relevant information retrieving from simultaneous messages. Thus, the problem has been modeled as predicting whether two messages come from the same-thread. While the previous study by (Jiang et al., 2018) seems to perform well on same-thread prediction, we find that it is because the data are randomly split into training and test sets, resulting overlapping of topics in training and testing sets. When data is split by time order, the performance of existing models drop significantly. In this study, we consider the problem of direct reply predication task and study different message pair classification models for the task. We argue that independent message encoders could better represent messages to capture their interaction than shared message encoders especially for direct-reply prediction task. We also find that BERT model performs well with small datasets, while other models may outperform BERT with large datasets.

1 Introduction

With the continuous development of the Internet and social media, online group discussions and conversations have become increasingly popular and play an important role in society and the economy. Many commercial sites take advantage of this to advertise their products or help solve users' problems. For example, developing conversational agents (chatbots) on their website to help staffs answer questions that may have been asked before.

The goal of chatlog disentanglement is to cluster messages that belong to the **same topic** for tracing. Selecting a reply for a given input or finding question-answer pairs are special cases of conversation disentanglement. For example, Figure 1 shows a segment of conversations consisting of four ongoing threads in the IRC (Internet Relay Chat) conversation. As we can see, interleaved conversations can occur in both two-person or multi-person chats. Thus, the goal of conversation disentanglement is to match the message pairs for question-answer pair generation.

There are many studies on conversation disentanglement in the past. One solution for conversation disentanglement is to model the topic of messages by estimating the similarity between messages and decide whether each incoming message starts a new topic or belong to an existing thread.

Jiang et al. (2018) proposed Siamese Hierarchical Convolutional Neural Network (SHCNN) which integrates two hierarchical CNNs to capturing low-order and high-order semantics of the messages for a better message representation. By concatenating absolute difference of the two representation with other temporal and user information, SHCNN predicts the probability of two messages belonging to the same-thread with high accuracy on the data of IRC and Reddit. However, the model performs poorly for future unseen messages when the training and testing data are split by time order.

We argue that the proposed data preparation method by (Jiang et al., 2018) only avoid the generation of too many negative examples, but may produce false positive message pairs as many subtopics fork from the main topic. When two messages from dif-

Thread	Speaker	Message
T77	Elli	Any idea why 'passwd' would ask for a new password four times?
T77	**Priscila**	Elli: A hacked version.
T78	Melda	is there a way to get ls -l to print full path in each response?
T75	Arlie	Julietta, do whatever you want ... its an ethernet packet
T71	Leota	Jeanice: i had to replace most of the the startup scripts with just echo boo, the problem seemed to be with the kernel not being detected or something
T78	Melda	so it'll show /home/user/filename.jpg at the end of every single line?
T77	Elli	Priscila, yeah that could be a possbility except I just recompiled it from sources, thinking just that
T77	Elli	And still the same behavious
T77	**Priscila**	Elli: Hmmm. Weird indeed ..
T75	Julietta	Arlie, well, i can't, to actually inject valid packets i would need to modify the sockets state
T71	Jeanice	Leota: strange... and what errors did you get when you tried to compile?
T78	**Priscila**	Melda: ls -l /home/user/*

Figure 1: Conversation in real-world chatting room

ferent subtopic are paired as positive examples, even humans have difficulty to recognize their relationship. In this paper, we consider a different task to pair only **direct-reply** messages for positive examples, synthesizing a more reasonable data set for question-answer pair extraction. We consider 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) and show improved performance over SHCNN. However, the best performance only achieves 0.669 F1, even with the BERT sentence pair classification.

In addition, we apply the direct-reply prediction task to extract question-answer pairs from chatlogs and find substantial labeling is required to obtain acceptable model. To speed up the process, we adopt heuristic labeling of the next sentence as a reply message to speed up training data preparation. Overall, chatlog disentanglement is still a challenging problem to be solved.

2 Related Work

The early research on conversation disentanglement can be traced back to the study on topic detection and tracking conducted in (Allan, 2002). As mentioned in (Shen et al., 2006), the messages in the same-thread have higher similarity. Thus, calculating message similarities based on linguistic features based on bag-of-words representation has been the major idea in (Elsner and Charniak, 2008). In addition, (Wang and Oard, 2009) showed that contexts can be used to improve the performance of message similarity calculations. However, overlapping contexts could also influence the calculation of similarity, leading to reduced performance.

Mehri and Carenini (2017) proposed a pipeline for the task of thread disentanglement, including reply classifier, same-thread classifier, next utterance classifier, and in-thread classifier. Of the three subtasks, only the third classifier, i.e. for "next utterance classification", can leverage unlabeled data to model message relationships to train an Recurrent Neural Network (RNN) classifier (Lowe et al., 2015). Finally, the "in-thread" classifier takes the output of the previous three classifiers "same-thread", "Reply", and "Next Utterance" to predict if an input message belongs to a thread.

To investigate how message similarity could be estimated, Jiang et al. (2018) proposed SHCNN (Siamese Hierarchical Convolutional Neural Networks) for the same-thread task prediction. They merged messages from different subReddits to simulate the concurrent conversations with multiple threads and generated a synthetic dataset of interleaved conversations, where messages from the same reddits within a limited elapsed time are paired to be positive examples, while messages from different reddits are paired to be negative examples. The experimental results show that the model performs well with 0.8392 MRR when the data are randomly

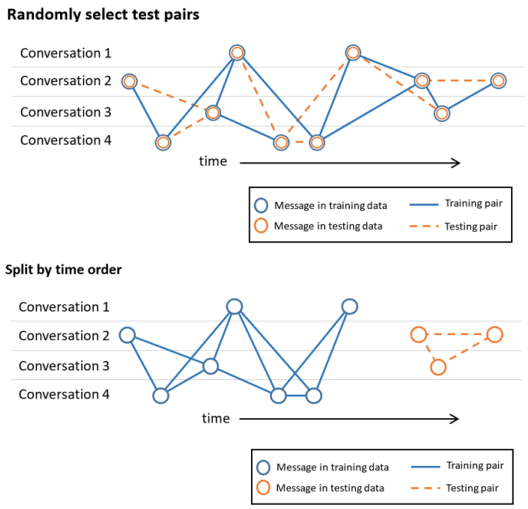

Figure 2: Split training data and testing data (a) randomly or (b) time order

split into training and testing sets as depicted in Figure 2(a). However, the performance of SHCNN model drops significantly when we use a time order splitting method as shown in Figure 2(b). In other words, the high performance reported in the paper may due to data peeping rather than a good model. In fact, since the messages in the same-thread could fork subtopics as the conversation goes on, it is difficult to judge whether two messages are related to each other directly, even for humans.

2.1 Message pair classification tasks

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

Aspect-level sentiment analysis aims to determine the sentiment polarity of a review sentence with respect to a given aspect. Many models and methods have been proposed from traditional machine learning methods (Schouten and Frasincar, 2016) to deep learning models (Zhou et al., 2019). For example, Wang et al. (Wang et al., 2016) proposed an attention-based LSTM network for aspect-level sentiment classification. Huang et al. (2018) introduced an attention-over-attention (AOA) neural network to capture the interaction between aspects and context sentences. The AOA model outperforms previous LSTM-based architectures. More models on aspect-level sentiment analysis can be found at (Zhou et al., 2019).

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.

2.2 Learning sentence representation

Note that most models mentioned above use pre-trained word embedding for the input layer and adopt RNN or CNN to learn sentence representation. Recently, learning sentence representation from large unlabeled corpus has become feasible. For example, Radford et al. Radford2018ImprovingLU suggested a two-stage training process called Generative Pre-Training (GPT). Devlin et al. (2018) improved GPT with Bidirectional Encoder Representations from Transformers (BERT), which also uses two-stage training process and stacked Transformer. Two tasks are considered in the pre-training stage, including masked language model (MLM) and next sentence prediction (NSP). Many NLP tasks built on top of BERT have been shown to surpass the previous state-of-the-art systems, including question answering and sentiment analysis (Sun et al., 2019).

3 Problem Definition and Datasets

Both the **same-thread** and **direct-reply prediction** tasks are binary classification problems with training data represented as (x,y), where x=(m_1,m_2) and $y \in \{0,1\}$ denoting whether m_1 and m_2 come from the same topic or m_2 is a reply of m_1.

There are two data sets used in this paper, namely IRC (Internet Relay Chat) and Reddit.

The IRC data set is a manually tagged data set used in (Elsner and Charniak, 2008). IRC provides group chats thus multiple conversations are interspersed with each other in a channel. This data set contains 6 hours of messages from the LINUX channel. Each message is annotated with the conversation or thread it is involved. Thus, it is consistent with the same-thread task.

The Reddit Dataset consists of comments from

Dataset	Reddit			IRC
	Gadgets	Iphone	Politic	
Conversations	468	529	6,197	159
Messages	11,071	10,261	148,942	1,865
Speakers	6,387	4,506	28,365	183
All pairs	487,695	507,226	4,492,361	79,682
same-thread pairs	118,889	111,145	1,226,863	5,390

Table 1: Datasets for the same-thread task

Dataset	Direct-Reply Prediction Task		
	Gadgets	Iphone	Politic
Conversations	18,220	34,348	27,361
Messages	358,212	345,487	800,619
Speakers	118,225	49,571	72,787
All pairs	1,143,058	947,484	2,423,900
Reply pairs	228,438	189,353	477,780

Table 2: Datasets for the direct-reply prediction task

Reddit articles which are synthesized by following Jiang et al.'s work (Jiang et al., 2018). Reddit is a web content rating and discussion website, where members can submit contents such as links or news or text posts which are then voted up or down. Posts are organized by subjects into user-created subreddit. Theoretically, all comments from the same post can be treated as the same-thread messages. Jiang et al. mix comments from different articles around the same time to create conversation logs of multiple people chat for the same-thread tasks, as shown in Figure 3.

Figure 3: Combination of Reddit dataset

Since the original posts are usually longer than comments, we keep only messages that are replies to comments and remove comments to the original articles such that most messages are about the same length. We collect three subreddits from gadgets, iPhone and politics channels from 2016/06 to 2017/05 and remove articles with too many comments and enumerate message pairs that are within T (one-hour) time span to prepare training data for the same-thread task as stated by Jiang, et al.

We follow (Jiang et al., 2018) to synthesize the dataset: For the direct-reply prediction task, every comment in Reddit is a reply to a previous message, which makes it a good source for the direct-reply prediction task. For this task, each comment is paired with five messages: with only one correct reply message and four other messages within time interval T (=1 hour), which may come from the same-thread or different threads. Table 1 and Table 2 show the number of conversations, messages, average messages per conversation, speakers, message pairs prepared for the same-thread task and the direct-reply prediction task, respectively. Because random splitting of data into training, validation and testing data may cause the message pairs from the same-thread to occur in both training and testing sets as shown in Figure 2(a), making the performance untrustworthy, we split the data based in chronological order (Figure 2(b)) to avoid data peeping, and use the first 72% of data for training, the following 8% for validation and the last 20% for testing.

4 Methods

In this paper, we consider models based on GloVe word embedding and BERT models for message-pair classification.

4.1 Glove-based Representation

A typical neural network model consists of embedding layer for word representation, hidden layer such as mutual attention for message representation, and output layer for prediction. For embedding layer, we adopt pre-trained GloVe (Pennington et

al., 2014) word embedding matrix from Common Crawl dataset (840B tokens), which contains a case-sensitive vocabulary of size 2.2 million. Given a message $m = [w_1, w_2, \ldots, w_L]$ with L tokens, we look up the embedding vector $u_i \in R^{d_w}$ for each word. If a token in the message does not exist in the pre-trained model, we replace it with the UNKNOWN token embedding. Thus, the message m is represented by a $L \times d_w$ matrix $\mathbf{U} = [u_1, u_2, \ldots, u_L]$.

We consider two models for message representation. The first one is GCNN-LSTM, and the second is LSTM with dual attention.

GCNN-LSTM Representation

Convolutional neural networks (CNN) are shift invariant artificial neural networks with shared-weights architecture and translation invariance characteristics, commonly applied in image processing. With d_c kernels of size $d_w \times k$, the output of the 1-D CNN layer will be $L \times d_c$ feature matrix. Here, we use Gated Linear Unit (GLU) proposed in (Dauphin et al., 2017) to control which information flows in the network.

$$\tilde{\mathbf{U}} = (\mathbf{U} * W + \mathbf{b}) \otimes \sigma(\mathbf{U} * W' + \mathbf{b}') \quad (1)$$

where $W, W' \in R^{k \times d_w \times d_c}$ and $\mathbf{b}, \mathbf{b}' \in R^{d_c}$ denote the parameters for two CNNs, one for feature extraction and one for GLU.

To deal with word sequence, we adopt a BiLSTM layer to capture the message information. LSTMs (Long Short-Term Memory) are recurrent neural networks that are able to capture ordered information from input sequence of tokens. BiLSTM is obtained by stacking two LSTM networks to get information from backwards and forward states simultaneously.

$$h_i = \overrightarrow{h_i} \oplus \overleftarrow{h_i} \quad (2)$$

Let θ_r denotes the parameters for BiLSTM, we define the output of BiLSTM function $G_r(\tilde{U}; \theta_r)$ as the sum of all hidden states, i.e.

$$G_r(\tilde{U}; \theta_r) = h_1 + h_2 + \ldots + h_L \quad (3)$$

Prediction and Objective Function

Let v_1 and v_2 denotes the output of the two input messages m_1 and m_2. For prediction, we use two fully connected layers to make the prediction, i.e.:

$$\hat{y} = \sigma(ELU([v_1, v_2]^T W_0 + b_0) W_1 + b_1) \quad (4)$$

where $W_0 \in R^{2d_r \times d_f}$, $W_1 \in R^{d_f \times 1}$. Let Θ denote the parameters used in the model to encode the sentence, i.e. $\Theta = \{W, W', b, b', \theta_r, W_0, b_0, W_1, b_1\}$, the model is trained by minimizing cross-entropy with L2 regularization as shown below.

$$\begin{aligned} Loss(D) \quad &= \textstyle\sum_{(x,y) \in D} y \cdot \log \hat{y} \\ &+ (1 - y) \cdot \log(1 - \hat{y}) + \lambda \left\| \Theta \right\|^2 \end{aligned} \quad (5)$$

LSTM-Dual Attention Model

Inspired by the power of attention mechanism, the second model we proposed is BiLSTM with dual attention.

Attention Mechanism is originally designed to help the decoder of seq2seq model generate words one by one. The idea is to collect the output vector h_i ($\in R^{2d_r}$) at each word for attention. Let $z_1 = G_r(\mathbf{U_1}; \theta_r)$ and $z_2 = G_r(\mathbf{U_2}; \theta_r)$, we can exploit attention mechanism to generate a representation for z_1 based on the content of z_2. Specifically, we calculate the weighted sum of the output at each step of the first message, $[h_1^1, h_2^1, \ldots, h_L^1]$ with weight vector α^1 as below:

$$z_1' = \sum_{i=1}^{L} \alpha_i^1 h_i^1 \quad (6)$$

where $\alpha^1 = [\alpha_1^1, \alpha_2^1, \ldots, \alpha_L^1]$ is computed by taking the inner product of z_2 with each h_i^1.

$$\alpha_i^1 = softmax(z_2^T h_i^1) = \frac{exp(z_2^T h_i^1)}{\sum_{j=1}^{L} exp(z_2^T h_j^1)} \quad (7)$$

Similarly, z_2' is calculated by the weighted sum with weight vector α^2. Finally, we concatenate z_1 with z_1' to form v_1 ($\in R^{4d_r}$), i.e. $v_1 = z_1 \oplus z_1'$, and z_2 with z_2', i.e. $v_2 = z_2 \oplus z_2'$, to form v_2 for classification task as described in Equation 4.

Attention over Attention (AOA) Model

For two output hidden states from BiLSTM $h_1 \in R^{n \times 2d_h}$ and $h_2 \in R^{m \times 2d_h}$, AOA (Huang et al., 2018) first calculates a pair-wise interaction matrix $I = h_1 \cdot h_2^T$, where the value of each entry I_{ij} represents the correlation of a word pair among the two input messages) and compute both column-wise softmax, $\alpha \in R^{n \times m}$ and row-wise softmax, $\beta \in R^{n \times m}$.

$$\alpha_{ij} = \frac{exp(I_{ij})}{\sum_i^n exp(I_{ij})}, \beta_{ij} = \frac{exp(I_{ij})}{\sum_j^m exp(I_{ij})}, \quad (8)$$

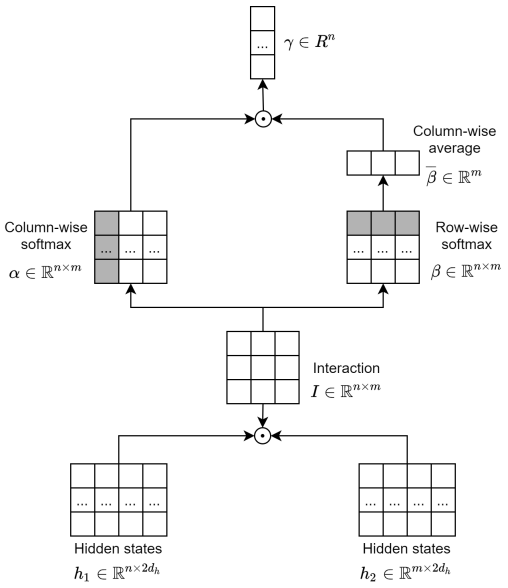

Figure 4: An Attention-over-Attention Module (AOA) (Huang et al., 2018)

AOA Layer

The idea of AOA is to compute the attention weight over the averaged attention weight $\overline{\beta} \in \mathbb{R}^m$ where $\gamma \in \mathbb{R}^n$,

$$AOA(h_1, h_2) = \gamma = \alpha \cdot \overline{\beta}^{\mathsf{T}}. \tag{9}$$

where

$$\overline{\beta}_j = \frac{1}{n} \sum_{i}^{n} \beta_{ij}. \tag{10}$$

We call γ the output of AOA layer and use it to calculate the final sentence representation $r \in R^{2d_h}$.

$$r(h_1, h_2) = h_1^{\mathsf{T}} \cdot \gamma \tag{11}$$

The final sentence representation r is then used for final result prediction, i.e. $\mathbf{p_o} = \mathbf{r}$.

$$P(y|x) = \sigma(\mathbf{w} \cdot \mathbf{p_o} + b_o) \tag{12}$$

The attention-over-attention layer structure is as shown in Figure 4.

4.2 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 same-thread/direct-reply 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 $([CLS], m_1, [SEP], m_2)$. Let H^l be the output of the transformer at layer l, thus $H^i = [h_0^l, h_1^l ... h_{n+m+2}^l]$ can be calculated by

$$H^{i+1} = BiTransformer(H^i), \tag{13}$$

The basic BERT sentence pair classification (BERT-SPC) model takes the output of [CLS] token as the prediction layer input, i.e. $\mathbf{p_o} = \mathbf{H_0^L}$ by Eq 12. The entire model is fine-tuned with a standard cross-entropy loss with L2 regularization.

BERT-SPC-AOA Model

To further improve BERT-SPC model, we concatenate the output $r(h_1, h_2)$ with [CLS] output as the input to the prediction layer, i.e. $\mathbf{p_o} = r(h_1, h_2) \oplus H_0^L$ by Eq. 12.

5 Experiments and Analysis

For non-BERT deep learning models, the pre-trained word embedding is GloVe (Pennington et al., 2014) with case distinction trained on the Common Crawl dataset to distinguish English word embedding, dimension d_w is 300 and total 2.2 million words. The hidden layers in BiLSTM h_r are 128, the number of kernels used in CNN h_c is 128, and kernel size k is 5. All models are implemented with Tensorflow. The batch size used in the traditional deep learning model is 256 and the maximum epoch and initial learning rate are set to 40 and $2 * 10^{-4}$.

For BERT model, the batch size is 32. 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. The L2 weight λ of the objective function in Equation 4 is set to 0.01. We apply linear attenuation to the learning rate and use warmup in the first 30% of the training step with dropout set to 0.1. For training data, the ratio of the positive versus negative (different thread) pairs is 1:1.

Dataset		Reddit						IRC	
		Gadgets		Iphone		Politic			
Measure		F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.
Rand. Split	SHCNN	0.779	0.887	0.608	0.816	0.639	0.770	0.805	0.967
	GCNN+LSTM	0.981	0.990	0.956	0.980	0.945	0.969	0.319	0.801
	LSTM+DualAtt	0.809	0.900	0.618	0.804	0.638	0.775	0.346	0.854
	AOA	0.979	0.990	0.933	0.970	0.812	0.887	0.571	0.915
	BERT SPC	**0.988**	**0.994**	**0.994**	**0.994**	**0.996**	**0.970**	**0.888**	**0.985**
Time Order Split	SHCNN	0.253	**0.709**	0.318	**0.553**	0.411	0.553	0.039	**0.939**
	GCNN+LSTM	0.289	0.645	0.364	0.516	0.309	0.601	0.098	0.630
	LSTM+DualAtt	0.308	0.710	0.417	0.500	0.441	0.616	0.089	0.854
	AOA	0.310	0.634	**0.428**	0.541	**0.482**	0.560	0.084	0.557
	BERT SPC	**0.522**	0.508	0.298	0.534	0.423	**0.667**	**0.223**	0.898

Table 3: Performance comparison of the same-thread prediction task.

Evaluation method

Because both tasks are modeled as binary classification problems and the numbers of same-thread pairs or direct-reply message pairs are fewer than different thread and non-reply message pairs, we consider them as positive data and use F1 and accuracy for the evaluation.

5.1 Same-thread Task

We start with the same-thread prediction task. Table 3 shows the performance of the models trained on a random or time order split data sets. As we can see, all the proposed models outperform SHCNN. GCNN-LSTM model and BERT SPC model perform especially good on random split data with 0.981 and 0.988 F1 on Reddit Gadgets dataset. However, the performance of all models drops significantly when data is split by time order. The F1 of the Reddit datasets nosedives from above 0.9 to 0.2 and 0.3 for GCNN+LSTM model. Even for BERT model, the plunge on IRC dataset is also precipitous (from 0.888 to 0.223).

In order to understand why all models perform poorly in the same-thread task, we conducted an error analysis and found the task to be challenging even for human beings. Table 4 shows some mislabeled message pairs and their ground truth labels. We notice that it is not easy to recognize the connections between the message pairs (e.g. the first three message pairs) that come from the same-thread without context. Meanwhile, annotators might be misled to give positive labels when two messages mention about the same entity, while the messages actually come from two different conversations. For example, the last message pairs both mentioned about Hillary, but the messages actually come from two threads. Thus, we argue that the same-thread task is a very difficult task when no context is given. However, context might not always help when multiple threads are mixed together as studied in (Wang and Oard, 2009).

To confirm our speculation, we randomly select 500 message pairs from Reddit test data, and give them to four graduate students to judge if each message pair comes from the same-thread. Only one out of 4 annotators is able to achieve higher than 0.5 F1 and the average F1 is only 0.340 (Table 6), indicating the difficulty of the same-thread task. In fact, it is not enough for annotators to rely on only two messages to determine whether they are from the same-thread. On the other hand, the performance could be greatly improved to 0.660 F1 and 0.883 accuracy in the direct-reply predication task.

5.2 Direct-reply Prediction Task

In view of the above problem, we consider the direct-reply prediction task and only split data based on time order to see how well models could perform for future application. As shown in Table 6, the performance of manual annotation on the direct-reply prediction task is much better than that for the same-thread task. Most models have performance higher than 0.5 F1 for the direct-reply prediction task.

As shown in Table 5, LSTM with dual attention

m_1	m_2	Label
He actually didn't shoot anyone who was walking down the street.	Agreed. A shallow grave in the woods is more fitting.	True
He teared up thinking about how this will make his re-election campaign more difficult.	House Freedom Caucus "You're free to die, you sick moochers. Isn't it beautiful!"	True
I wonder if he used all the best words?	They can't have my brand!	True
Trump tries to clean up on Whitewash Crimea	Trump is a traitor and will sell this country out to the highest bidder the moment he gets into office.	False
Pepperidge Farms remembers people saying Hillary "Warmonger" Clinton.	I love how he framed Hillary as the warhawk.	False

Table 4: Some mislabeled examples and their true labels for the same-thread task.

direct-reply Task	Gadgets		Iphone		Politic	
	F1	Acc	F1	Acc	F1	Acc
SHCNN	0.513	0.845	0.455	0.849	0.512	0.849
GCNN+LSTM	0.591	0.856	0.534	0.845	0.573	0.861
LSTM+DualAtt	0.598	0.869	0.567	0.856	**0.637**	**0.870**
AOA	0.514	0.721	0.486	0.723	0.566	0.778
BERT SPC	0.632	**0.883**	0.543	**0.862**	0.616	0.818
BERT-SPC-AOA	**0.669**	0.877	**0.617**	0.840	0.623	0.825

Table 5: Model performance for the direct-reply prediction task

	same-thread		direct-reply Prediction	
Measure	F1	Acc	F1	Acc
Annotator 1	0.105	0.728	0.548	0.860
Annotator 2	0.352	0.763	0.800	0.930
Annotator 3	0.263	0.732	0.590	0.875
Annotator 4	0.638	0.796	0.703	0.865
average	0.340	0.755	0.660	0.883

Table 6: Performance of four annotators for the same-thread and direct-reply prediction task.

outperforms BERT on two of the Reddit datasets except for Gadgets. However, the best performance on Gadget Reddit dataset is achieved by BERT-SPC-AOA model. Compared with the best result (0.522, 0.428 and 0.482 F1) for the same-thread prediction task, We can see the performance on three Reddit datasets is improved to 0.669, 0.617, and 0.637 F1 by BERT-SPC-AOA and LSTM with dual attention model.

6 Conclusion

This paper addresses the rationality of the chatlog disentanglement problem in two ways. First, the testing data should be prepared to simulate future unseen data. Second, the same-thread task without context information is too challenging even for human annotators. Thus, we propose the direct-reply prediction task for question-answer pair generation from chatlogs. In the direct-reply prediction task, using the pre-trained BERT model for Fine-Tuning can achieve good performance, even with less training data. However, the data quality used by the downstream task seems to be a big problem when using BERT for Fine-Tuning. When a large number of messages are disentangled, the negative examples for both the same-thread or direct-reply prediction tasks are much larger than the positive examples. Though, down sampling is adopted to balance the training data, the models still tend to classify the message pairs to the negative session, leading to low F1.

For future work, how to add context information is an alternative direction to consider. Meanwhile, as direct-reply predication is not a symmetric problem, an input of (m_1, m_2) is different from (m_2, m_1). Thus, we wonder that the shared message representation and the predication function of multi-layer

perception might not be enough to extract the features from two message representation. Thus, we might consider asymmetric models, i.e. two message encoders for each of the input messages to see if it could achieve better performance.

Acknowledgments

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

References

James Allan, 2002. *Introduction to Topic Detection and Tracking*, pages 1–16. Springer, Boston, MA.

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. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

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.

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.

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.

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.

K. Schouten and F. Frasincar. 2016. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 35–42, New York, NY, USA. Association for Computing Machinery.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. 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 380–385, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Lidan Wang and Douglas W. Oard. 2009. Context-based message expansion for disentanglement of interleaved text conversations. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, page 200–208, USA. 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.

Jie Zhou, Jimmy Xiangji Huang, Qin Chen, Qinmin Vivian Hu, Tingting Wang, and Liang He. 2019. Deep learning for aspect-level sentiment classification: Survey, vision, and challenges. *IEEE Access*, 7:78454–78483.