Journal of Information Processing Vol.27 742–751 (Nov. 2019)

[DOI: 10.2197/ipsjjip.27.742]

Regular Paper

Attitude Detection for One-Round Conversation: Jointly Extracting Target-Polarity Pairs

Zhaohao Zeng[1,a)] Ruihua Song[2,b)] Pingping Lin[2,c)] Tetsuya Sakai[1,d)]

Received: March 9, 2019, Accepted: July 3, 2019

Abstract: We tackle Attitude Detection, which we define as the task of extracting the replier's attitude, i.e., a target-polarity pair, from a given one-round conversation. While previous studies considered Target Extraction and Polarity Classification separately,we regard them as subtasks of Attitude Detection. Our experimental results show that treating the two subtasks independently is not the optimal solution for Attitude Detection, as achieving high performance in each subtask is not sufficient for obtaining correct target-polarity pairs. Our jointly trained model AD-NET substantially outperforms the separately trained models by alleviating the target-polarity mismatch problem. By employing pointer networks to consider the target extraction task a boundary prediction problem instead of a sequence labelling problem, the model obtained better performance and faster training/inference than LSTM and LSTM-CRF based models. Moreover, we proposed a method utilising the attitude detection model to improve retrieval-based chatbots by re-ranking the response candidates with attitude features. Human evaluation indicates that with attitude detection integrated, the new responses to the sampled queries are statistically significantly more consistent, coherent, engaging and informative than the original ones obtained from a commercial chatbot.

Keywords: conversation, chatbot, sentiment analysis, attitude detection

1. Introduction

Research in developing a chatbot has been active due to the large amount of conversations available on the Internet and recent progress made by neural network models [1], [2], [3]. However, no matter whether retrieval-based or generation-based, it is difficult for a chatbot to control the responses of a chatbot, and to give it a consistent personality. Several existing studies [4], [5], [6] attempted to attach personalities to chatbots to make them more consistent and engaging, but the form and the definition of "personality" are not entirely clear. In the PersonaChat study [6], the annotators were asked to compose personality profiles to depict some specific personalities, and each personality profile consists of multiple sentences of textual description, e.g., "*I am a vegetarian. I like swimming. My father used to work for Ford. My favourite band is Maroon5.*....." After inspecting the personality profiles in PersonaChat training dataset, we found that almost half of sentences in them are possibly expressing attitudes. We define an *attitude* as a target-polarity pair, where the target is an entity explicitly discussed in a sentence, and the polarity is a sentiment expressed towards that target, which can be either positive, negative, or neutral. For example, the sentence "*My favourite band is Maroon5*" expresses such an attitude: positive → Maroon5. In the PersonaChat dataset, 92.8% personality profiles and 49.2% sentences of personality profiles contain at least one sentiment word [*2] (like, enjoy, hate, etc.). In the present study [*3], we consider attitude an important factor in personailising chatbots, and hypothesise that enabling chatbots to express appropriate attitudes can also make conversations more consistent. Current chatbots may express attitudes without any consistency, as shown in **Table 1**. When a chatbot shows contradictory sentiments towards the same target, the user may be confused, and it may hurt user experience.

Table 1 Inconsistent responses with different sentiment polarities towards "stay up late". These responses were collected from Microsoft Xiaoice [*1], a Chinese commercial chatbot, and manually translated into English.

query	Do you like staying up late?	
response	I don't do it anymore since I have to work on the next day.	(Negative)
query	Do you also like staying up late to watch?	
response	I don't like it.	(Negative)
query	Do you also like staying up late?	
response	I don't stay up late.	(Negative)
query	Don't you like to stay up late?	
response	I like staying up late, and I don't like getting up early.	(Positive)
query	I want to stay up late	
response	It's more health to stay up late.	(Positive)

In the present study, we propose a method which can be applied to an existing chatbot system to control its attitudes and subsequently make the chatbot converse in a more consistent and coherent manner. Also, we suppose controlling attitudes may

[1] Waseda University, Shinjuku, Tokyo 169–8555, Japan
[2] Microsoft, P.R. China
[a)] zhaohao@fuji.waseda.jp
[b)] song.ruihua@microsoft.com
[c)] pingping.lin@microsoft.com
[d)] tetsuyasakai@acm.org

[*1] https://www.msxiaobing.com
[*2] Sentiment word list: www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon
[*3] An early version of this paper was presented at ACM WSDM 2019 [7].

contribute to shaping personalities for chatbots, which may eventually lead to more engaging and informative conversations. In brief, we assume that for a user query, a chatbot system can obtain a list of ranked response candidates by searching sentences from a corpus (retrieval-based) or from the language model of its decoder (generation-based). Then, given an attitude profile which consists of the desired attitudes (e.g., Negative → stay up late) that the chatbot should hold, the response candidates are re-ranked based on the attitudes extracted from the query and each response candidate.

To automatically detect the attitude expressed by each response candidate given the user query, we attempt to tackle Attitude Detection, which we define as the task of extracting the replier's attitude (i.e., a target-polarity pair) from a given one-round conversation (i.e., a query-response pair). As an attitude is defined as a target-polarity pair in the present study, the task Attitude Detection (AD) consists of two subtasks: Target Extraction (TE), which identifies the attitude target from the text, and Polarity Classification (PC), which classifies which sentiment polarity (positive, neutral or negative) the attitude holds. In the present study, we extract a target as a span of text from either the query or the response, i.e., explicit target [8], and it does not require a domain-specific list of pre-defined targets. Moreover, most existing studies about Attitude Detection and its subtasks focus on sentence-level and document-level data, but the attitude contained in a conversation utterance may not be detected easily by neither sentence-level models nor document-level models due to the need of context information in the conversation. For example, in the first conversation of Table 1, it is impossible to find the target by looking at the response utterance alone as the target is only mentioned in the query. Hence, we propose AD-NET which can extract the replier's attitude by looking at both query and response.

Although the TE and PC subtasks of Attitude Detection are traditionally trained and evaluated in isolation [9], we propose AD-NET to be jointly trained for both TE and PC subtasks in an end-to-end manner. As our first contribution for attitude detection, we evaluate AD-NET with 22,000 human-human Chinese one-round conversations. We demonstrate that treating the two subtasks independently is not the optimal solution because achieving high performance in both subtasks separately will not be sufficient to obtain high performance in Attitude Detection where target and polarity are evaluated as a pair. In our experiment, the jointly trained model AD-NET can alleviate the target-polarity mismatch problem, and it outperforms the separately trained models in terms of attitude detection. In addition, by employing pointer networks to consider the target extracting task a boundary prediction problem, the model obtained better performance and faster training/inference than LSTM and LSTM-CRF based models.

To verify the approach to re-ranking response candidates with attitudes, we build an attitude profile manually, and use its attitude targets as keywords to sample queries from the user log of Microsoft Xiaoice (a Chinese commercial chatbot) . Human evaluation indicates that with attitude detection integrated, the new responses to the sampled queries are statistically significantly more consistent, coherent, engaging and informative than the original ones obtained from a commercial chatbot.

2. Related Work

2.1 Attitude Detection

The present study is inspired by Li et al. [9]. They use memory networks to classify if a target candidate is the attitude target of the input sentence, and then output the corresponding sentiment to the target candidate in an end-to-end manner. They consider Target Extraction a binary classification problem, so their method feeds each possible target into the model and perform a binary classification to determine which target was involved in the sentence. This method is acceptable for datasets which have very few distinct targets (e.g., there are only 5 possible targets in their experiment for Twitter stance detecting), it may not be directly applicable to open-domain applications, where a lot of entities are potential targets, to perform thousands times of binary classification for each sentence. In addition, the target candidate list is a finite set of nouns, so it cannot recognise unseen targets. To address these limitations, our AD-NET model extracts targets from the given text and therefore does not require a list of target candidates.

2.2 Target-Level Sentiment Classification

Target-level sentiment classification is a task where the system is only required to identify the polarity of a manually annotated target. Traditional approaches for this task include rule based [10] and statistical based methods [11], which require handcrafted features. Recurrent Neural Networks (RNNs) based models have the ability to predict labels by learning appropriate representations from raw text, and have been widely used in this task [12], [13]. Recently, memory networks have been proven to work well for this problem. For example, Tang et al. [14] proposed to use multi-hop memory network on this task and achieved competitive performance. Chen et al. [15] adopt multiple-attention mechanism to improve the performance of the memory network model.

2.3 Explicit Target Extraction

As Target Extraction can be considered a sequence-labelling problem, several recent studies applied Conditional Random Fields (CRFs) with handcrafted features to this task [16], [17]. More recently, neural network based CRFs have been proposed to improve the manually-designed features. Poria et al. [18] improved linear CRFs models by using convolutional neural networks to obtain non-linear features for the emission scores of CRFs. Wang et al. [19] proposed to incorporate recursive neural networks and CRFs to learn high-level discriminative features. In contrast to these sequence labelling methods, our model outperforms LSTM and LSTM-CRF based models by treating target extraction as a boundary prediction problem.

2.4 Context-Dependent Sentiment Analysis

While some researchers have investigated the problem of sentiment analysis with contextual information, most of these studies are about sentiment polarity classification instead of target extraction. Vanzo et al. [20] proposed SVM^{hmm} with manually designed context-dependent features to assign sentiment polarities to a stream of tweets. CRFs are also utilised by consider-

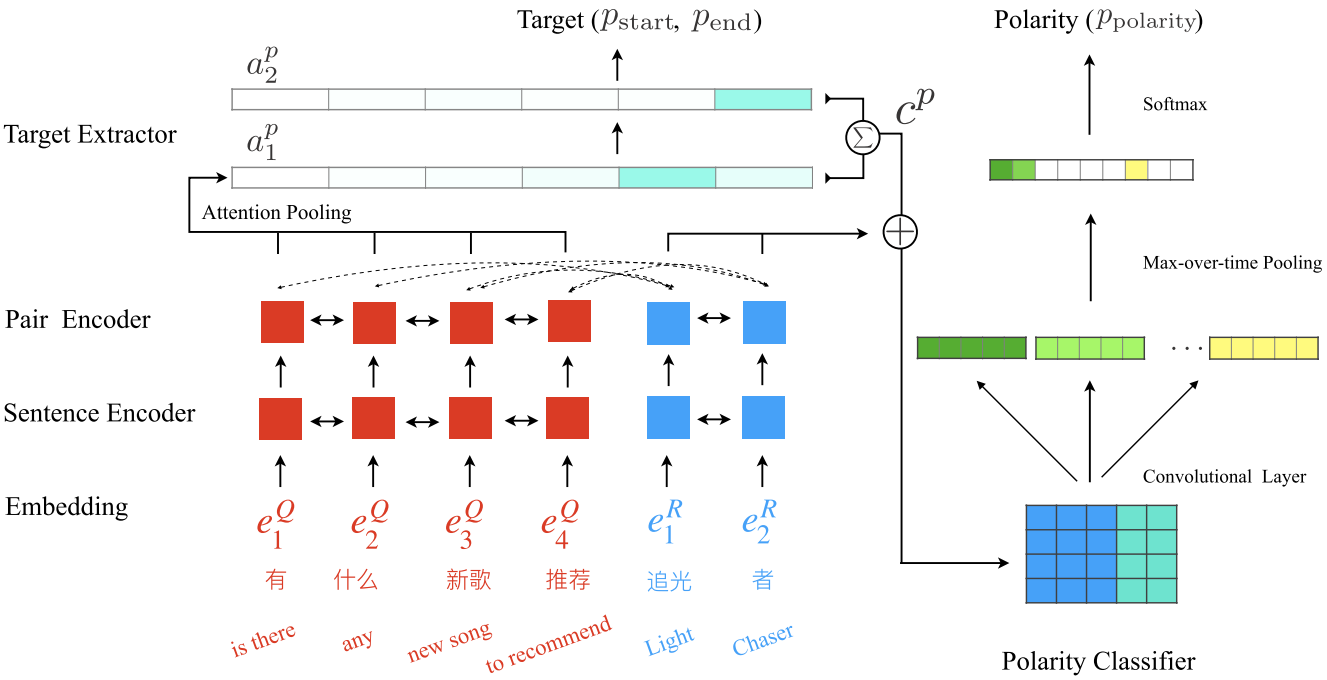

Fig. 1 The Architecture of AD-NET. The red part is for modeling the query and the blue part is for modelling the response.

ing assigning polarities to a stream of tweets as a sequence labelling problem [21], [22]. Feng et al. [23] introduced a hierarchical attention based LSTM model to assign polarity to a tweet with contextual information from its preceding tweets. Instead of a thread of tweets, our proposed model focuses on the query-response pair and utilises pair encoding to model the interactions between queries and responses in conversations. Moreover, our method not only classifies sentiments but also extracts targets.

2.5 Personailising Chatbots

To assign consistent personalities to chatbots, Li et al. [4] proposed to provide the speaker and user embeddings to the decoder of sequence-to-sequence models, and the models are trained with Twitter Persona dataset and television series transcripts. Instead, Qian et al. [5] proposed to feed pre-defined personality/identity attributes to the decoder. In their study, a personality profile is a list of key-value pairs, and is mainly about identification like name, age, and gender. Zhang et al. [6] released the PersonaChat dataset, which consists of personality profiles written by crowdworkers and the corresponding conversations. Each personality profile consists of multiple sentences of textual description. They proposed to use attention mechanism to obtain personality features from the personality profiles during sequence-to-sequence decoding step in their experiments. These three studies focused on generation-based chatbot models, and their chatbot models need to be trained on special training datasets (i.e., must have the corresponding personalities for each dialogue). In contrast, our method only requires to re-rank the generated response candidates, and can be integrated into any trained generation-based and retrieval-based chatbots.

3. Methodology

3.1 Attitude Detection

Given a one-round conversation $\{Q, R\}$, where the query $Q = \{w_t^Q\}_{t=1}^m$, the response $R = \{w_t^R\}_{t=1}^n$, $w_t^Q \in \mathbb{R}^{V\times 1}$ denotes the t-th one-hot word vector of the query, $w_t^R \in \mathbb{R}^{V\times 1}$ denotes the t-th one-hot word vector of the response, and V denotes the vocabulary size. In addition, $\{w_t^D\}_{t=1}^{m+n}$ denotes the concatenation of Q and R. For each conversation, the goal of attitude detection is to extract the target $T = (w_{p_{start}}^D, \ldots, w_{p_{end}}^D)$, which is a text span of either the query or the response, and identify the corresponding sentiment polarity $p_{polarity} \in \{\text{Positive}, \text{Neutral}, \text{Negative}\}$.

Figure 1 shows the architecture of the proposed model. Specifically, our model encodes the contextual information of query and the response of the conversation using RNNs. Attention-based recurrent networks are employed to symmetrically obtain response-aware representation for the query and query-aware representation for the response. Then, the boundary of the target span is predicted by the target extractor. The intermediate target vector is concatenated to the representation of the response, and then fed to a target-aware classifier to predict the polarity. The whole model can be trained jointly in an end-to-end manner.

3.1.1 Sentence Encoder and Pair Encoder

We use an embedding matrix A to convert one-hot word representation $\{w_t^Q\}_{t=1}^m$ and $\{w_t^R\}_{t=1}^n$ into distributed representation $\{e_t^Q\}_{t=1}^m$ and $\{e_t^R\}_{t=1}^n$. Then, stacked LSTMs encode each word in the query and the response to obtain the new representations $\{h_t^Q\}_{t=1}^m$ and $\{h_t^R\}_{t=1}^n$ respectively.

$$h_t^Q = \text{LSTM}^Q(h_{t-1}^Q, e_t^Q), \qquad h_t^R = \text{LSTM}^R(h_{t-1}^R, e_t^R)$$

To model the query-response relationship, we use attention based recurrent networks to encode them again with informa-

tion from each other. This encoder was originally proposed by Rocktäschel [24] for entailment recognition. In our setting, it encodes information in an utterance again with attention to the other utterance, so we obtain the query-aware response representation and the response-aware query representation as follows:

$$u_t^Q = \text{LSTM}^Q(u_{t-1}^Q, [h_t^Q, c_t^R]), \quad u_t^R = \text{LSTM}^R(u_{t-1}^R, [h_t^R, c_t^Q])$$

where $c_t^R = \text{attention}(h^R, [h_t^Q, u_{t-1}^Q])$ is the context vector of the response representation h^R and $[\bullet, \bullet]$ denotes concatenation of vectors. Specifically, the context vector c_t^R, and the corresponding attention score s_t^R and normalised score a_t^R can be calculated as follows: $s_{t,i}^R = v_r^T \tanh(W^R h_i^R + W^Q[h_t^Q, u_{t-1}^Q])$ where $a_{t,i}^R = \text{softmax}_i(s_{t,i}^R)$, and $c_t^R = \sum_{i=1}^n a_{t,i}^R h_i^R$. The context vector of the query representation c_t^Q can be obtained symmetrically from h^Q in the same way.

3.1.2 Target Extractor

To locate the target text span, we adopt the method widely used in Reading Comprehension [25], [26], which uses pointer networks [27] to predict the target span. That is, we use attention mechanism as a pointer to predict the start position p_{start} and the end position p_{end} from either the encoded query representation $\{u_t^Q\}_{t=1}^m$ or the response representation $\{u_t^R\}_{t=1}^n$. To do so, we concatenate their representations along the time dimension: $\{u_t^D\}_{t=1}^{m+n} = \{u_t^Q\}_{t=1}^m + \{u_t^R\}_{t=1}^n$ where $+$ denotes concatenation of sequences. Then, the attention mechanism is formulated as $s_{t,i}^p = v_p^T \tanh(W^D u_i^D + W^p h_{t-1}^p)$ where $a_{t,i}^p = \text{softmax}_i(s_{t,i}^p)$, $c_t^p = \sum_{i=1}^{m+n}(a_{t,i}^p u_i^D)$, c_t^p is the context vector obtained from attention mechanism, s_t^p is the attention score, and a_t^p is the normalised attention score. The pointer network can be considered a decoder of a sequence-to-sequence network which only decodes for two time steps, and the boundary of the target span (p_{start}, p_{end}) can be obtained from the normalised attention scores a_t^p during the two-step decoding. As there are only two steps, we employ Gated Recurrent Unit (GRU) instead of LSTM as $h_t^p = \text{GRU}(h_{t-1}^p, c_t^p)$ where $(p_{start}, p_{end}) = \arg\max_{i,j}(a_{1,i}^p \cdot a_{2,j}^p)$. We also apply attention pooling to the query representation to initialise $h_0^p = \sum_{i=1}^m a_{t,i}^g u_i^Q$, where $s_{t,i}^g = v_g^T \tanh(W^g u_i^Q + W^b)$, $a_{t,i}^g = \text{softmax}_i(s_{t,i}^g)$.

3.1.3 Polarity Classifier

To make the polarity classification conditional on the predicted target in an end-to-end manner, we utilise the context vectors (c_1^p, c_2^p) which are obtained from the target extractor. Ideally, the two context vectors should be close to the representation of the predicted target as they are the boundary words of the target. We concatenate the target representation $c^p = c_1^p + c_2^p$ to each time step of $\{u^D\}_{t=1}^{m+n}$, and utilise Convolutional Neural Networks (CNNs) [28] to classify the corresponding polarity. Specifically, we consider $\{[u^D, c^p]\}_{t=1}^{m+n}$ an image tensor which only has one channel. For each convolution filter, its width equals the dimensionality of the representation vector $[u_t^D, c^p]$, its height k is a hyper-parameter which represents an n-gram window size, and the feature map it produced is $\{z\}_{t=1}^{m+n-k+1}$. Then, max-over-time pooling is followed to obtain the feature of this filter. After the features from each filter are concatenated to produce the final feature $\hat{z}$, a fully connected layer and softmax are utilised to predict polarity $p_{polarity} = \arg\max_i(y_i^o)$, where $y^o = \text{softmax}(W^o \hat{z} + b^o)$.

3.1.4 Model Training and Inference

We utilise Cross Entropy as the loss function to jointly train the proposed model. For example, if the labels of the i-th training example are $(\hat{p}_{polarity}, \hat{p}_{start}, \hat{p}_{end})$, and then the training loss of this example will be $\mathcal{L}_i = -\log(a_{1,\hat{p}_{start}}^p \cdot a_{2,\hat{p}_{end}}^p \cdot y_{\hat{p}_{polarity}}^o)$. Note that the ground truth of the target span position $(\hat{p}_{start}, \hat{p}_{end})$ will be $(0, 0)$ (i.e., position of the start symbol) if the target is blank.

3.1.5 Model Variants

As Tang et al. [14] proposed that syntactic features may help sentiment-related models learn and generalise better, we try to add syntactic features to the AD-NET model. Specifically, we utilise a library called Language Technology Platform [29] to perform Part-of-Speech (POS) tagging, dependency parsing, and Named Entity Recognition (NER) for each utterance. For each token, we have the POS embedding and NER embedding using LTP. Also, the word embedding of its parent and the corresponding dependency relation embedding can be obtained from dependency parsing. We concatenate these features to the word embedding, and we refer to this variant as AD-NET + syntactic features.

Furthermore, To explore the usefulness of joint training, we modify AD-NET to make the polarity classifier independent from the target extractor. Specifically, the target extractor and the polarity classifier have their own sentence encoder and pair encoder, and the target representation c will not be concatenated to the representation of the query u^Q. Since these two parts are trained separately, we refer to this variant of our proposed model as *AD-NET-separate*.

3.2 Re-ranking with Attitudes

In re-ranking, the score s of each response candidate is mapped to a new score s' according to the attitude it expresses. First, an *attitude profile* is manually crafted to determine what attitudes the chatbot should hold. An attitude profile AP is a dictionary whose keys AP_{keys} are the targets and the values are the corresponding polarities. Then, given a user query Q and a response candidate R, the attitude target T and polarity P of the detected attitude are obtained from an attitude detection model. By comparing the extracted attitude with the items in the attitude profile, the responses candidate is categorised into three types:

Accordant The attitude profile contains the same attitude target and polarity: $T \in AP_{keys}, P = AP(T)$.

Neutral The attitude profile does not contain the same target: $T \notin AP_{keys}$ or there is no attitude expressed by $\{Q, R\}$.

Contradictory The attitude profile contains the same attitude target but its polarity is different from the one the response expresses: $T \in AP_{keys}, P \neq AP(T)$.

For example, an accordant response and a contradictory response are shown in **Fig. 2**. After obtaining the category, the new score s' of the response is adjusted by $g(s)$ or $g'(s)$:

$$s' = \begin{cases} g(s), & \text{if accordant;} \\ s, & \text{if neutral;} \\ g'(s) & \text{if contradictory.} \end{cases} \quad (1)$$

The choices of $g(s)$ and $g'(s)$ in our implementation are detailed in Section 4.2.1.

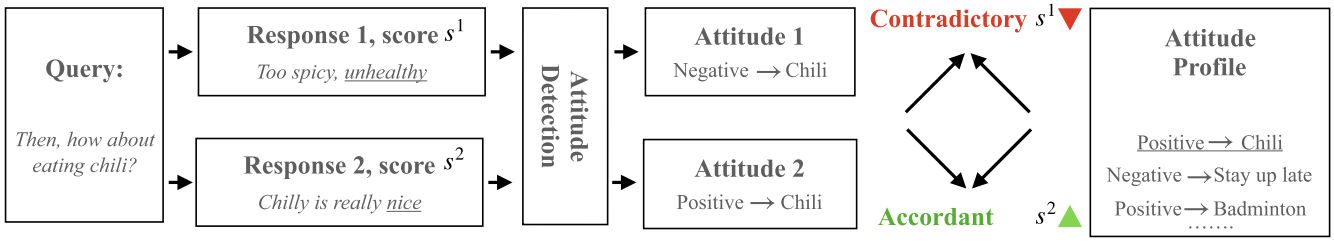

Fig. 2 Example of re-ranking with attitudes. If a response is accordant with the attitude profile, its score will be increased from s to $g(s)$. If a response is contradictory to the attitude profile, its score will be decreased from s to $g'(s)$. The attitude profile is manually crafted to configure what attitudes we want the chatbot to express in conversations. For evaluation, we also use the attitude targets (e.g., chili) in the attitude profile as keywords to sample user queries from the user log of a commercial chatbot.

4. Experiment

4.1 Attitude Detection

4.1.1 Dataset

To train and evaluate attitude detection models, we sampled 22,000 Chinese human-to-human one-round conversation data crawled from Chinese social network platforms, including Weibo [*4], and Douban [*5]. We split the whole dataset into three parts: training set (80%), development set (10%), and test set (10%). The statistics of the dataset are shown in **Table 2**. We hired 16 experienced annotators to provide attitude annotations. Specifically, each conversation in the training set was annotated by one annotator, and each conversation in the development and test sets was annotated by three annotators independently. Given a conversation, each annotator first read the query and the response, and then he/she selected the target text span and chose a corresponding polarity. If there was no attitude target, the annotator would just leave the target blank. Some examples of annotated dialogues are shown in **Fig. 3**. To clean the dataset, we removed 1) 243 (1.10%) examples because their annotated targets cannot be extracted as tokens after segmentation. 2) 557 (2.52%) examples because they have multiple annotated non-neutral attitudes.

4.1.2 Evaluation Metrics

While existing studies and contests consider target and polarity separately [30], [31], [32], we evaluate them together as attitudes, i.e., target-polarity pairs, to precisely understand the performance of models in the scenario of controlling a chatbot's attitudes. On the other hand, we treat a detected target-polarity pair as *correct* if it matches with the decision of at least one annotator. This is based on our view that every assessor's subjective decision is reasonable and that the model should accommodate diverse users rather than the average user. For conversations whose annotations are all neutral, the predictions will be correct if the predictions are also neutral. For example, if the annotations of Fig. 3 (c) consist of a positive and two neutral annotations, a correct prediction will have to be positive. However, if all three annotations are neutral, then a correct prediction will have to be neutral as well.

As there is a substantial imbalance between neutral and non-neutral polarities in our data, we compute an arithmetic mean of two F1 scores as described below. First, we define positive precision and positive recall as:

[*4] https://www.weibo.com
[*5] https://www.douban.com

Fig. 3 Examples of annotated dialogues. The attitude targets are underlined in the dialogues, and the corresponding polarities are shown in the captions.

Table 2 Statistics of the dataset for attitude detection.

Language	Chinese
Avg. query length (#tokens)	7.20
Avg. response length (#tokens)	7.51
#Conversations in Train	17,600
#Conversations in Dev	2,200
#Conversations in Test	2,200
#Neutral Conversations	13,961
#Positive Conversations	6,417
#Negative Conversations	1,622
#Annotators/Conversation in Train	1
#Annotators/Conversation in Dev&Test	3

$$P^+ = \frac{\#\text{Correct Positive Predictions}}{\#\text{Positive Predictions}}$$

$$R^+ = \frac{\#\text{Correct Positive Predictions}}{\#\text{Conversations with Positive Labels}}$$

where a positive prediction is considered correct if the extracted target exactly matches an annotated target *and* the polarities are also both positive. Negative precision and negative recall (P^- and R^-) are defined similarly. The positive and negative F_1 scores are given by $F_1^+ = 2\frac{P^+ \cdot R^+}{P^+ + R^+}$, $F_1^- = 2\frac{P^- \cdot R^-}{P^- + R^-}$, and the overall F1 is $F_1 = (F_1^+ + F_1^-)/2$.

4.1.2.1 Target Extraction

We evaluate target extraction in isolation by ignoring the polarity classification part, using F_1. An extracted target is correct if it exactly matches with one of the annotated targets. If all annotations are blank, an extracted target is correct if it is also blank. Since only targets are involved, we do not score them separately for positive conversations and negative conversations in the eval-

uation of target extraction.

4.1.2.2 Polarity Classification

We also evaluate polarity classification in isolation by ignoring the target extraction part, using overall F_1. A polarity is correct if it matches with one of the annotated polarities. If all annotations are neutral, a polarity is correct if it is also neutral. Similar to Attitude Detection, positive F_1^+ score and negative F_1^- score are computed separately and then overall F_1 is an arithmetic mean of them.

4.1.2.3 Human Performance

We measure human performance on our development and test sets. As we have three annotators for each conversation, we treat the first annotation as the human prediction, and keep the other answers as ground truths.

4.1.3 Baseline Models

We compare *AD-NET* with several baseline models. These baseline models are similar to *AD-NET-separate* which addresses two subtasks Target Extraction and Polarity Classification independently. The extracted targets and polarities are then combined for evaluation. The target extraction subtask is usually tackled as a sequence labelling task. We compare our model to the following BiLSTM and BiLSTM-CRF models. **BiLSTM**: A softmax layer is applied after the sentence encoder and the pair encoder to predict a traditional O-B-I-E tag (O: Not a target, B: beginning of a target, I: inside a target, or E: end of a target) for each token. **BiLSTM-CRF**: Similar to BiLSTM, but it utilises a stateful CRF loss layer to take the transition between different tags into account. CRF uses the forward-backward algorithm for loss calculation and the Viterbi decoding algorithm for inference, so it results in slower training and inference speed. For the Polarity Classification subtask, we implement the following state-of-the-art model **BiLSTM-Attn** [33], the top performer from SemEval 2017 task 4: Message-Level Sentiment Analysis. It applies attention pooling to the output of the pair encoder to obtain a dense vector for each conversation before the softmax layer.

While Li et al. [9]'s work is highly related to the present study, we choose not to implement their method as a baseline because their method requires to sequentially perform binary classification on a list of possible targets for each sentence. It is acceptable for the dataset in their study (e.g., there are only 5 possible targets in their tweet stance detecting dataset), but it may be inefficient for open-domain conversational data: we have 3,542 distinct targets in our training set, and performing 3,542 times of binary classification for each conversation may take too much time.

4.1.4 Hyperparamters

For word embedding, we pre-trained embedding vectors on 20 Gigabytes (i.e., five billion Chinese characters) conversational data from Douban. The embedding vectors have 100 dimensions and were trained using word2vec [34]. We froze the embedding vectors during training and use zero vectors to represent unknown words, start and end of sentences, and padding elements. During training, we added Gaussian noise ($\mu = 0, \sigma = 0.2$) to the embedding vectors, and dropout (rate = 0.3) is also applied to them. Instead of vanilla LSTM, we stacked two layers of Bidirectional LSTM (BiLSTM) for the sentence encoder and one layer of BiLSTM for the pair encoder. The size of hidden units was set to 256 for all layers. The dropout rate between LSTM layers was set to 0.5. The model was optimised by ADAM [35] with 64 batch size and 1×10^{-3} learning rate.

4.2 Re-ranking with Attitudes

4.2.1 Implementation of Re-ranking

As we discussed in Section 3.2, we need to define $g(s)$ and $g'(s)$ to adjust scores according to responses' attitude in re-ranking. To maximise the difference caused by re-ranking, we define $g(s) = s + 10^{12}$ and $g'(s) = s - 10^{12}$ in our implementation to make the accordant responses top the list. Furthermore, to avoid hurting relevance, we only use the top k response candidates for re-ranking, where $k = 80$ in the experiment [*6].

4.2.2 Data Collection

To verify if re-ranking with attitudes can retrieve better responses, we sample user queries from the user log (timestamp between Jun 1, 2018 and Jun 31, 2018) of Microsoft Xiaoice and compare their original retrieved responses with the re-ranked responses. We manually build an attitude profile *AP*, which contains about 100 attitudes and is used to shape a character (e.g., A boy likes chili, playing badminton, and dislikes staying up late). We treat the targets of attitudes in the profile as keywords to sample queries from the user log. In other words, each sampled query Q must satisfy $\{Q\} \cap AP_{keys} \neq \emptyset$. The reason is that if a query does not contain any attitude target, then its response candidates are unlikely to have the contradictory or accordant attitudes for re-ranking like Fig. 2.

Intuitively, if a keyword is less controversial, the response candidates in the corpus will be more likely to have consistent attitudes. For example, most people like traveling and most answer candidates about traveling also express positive attitude to it, then the re-ranking approach may be less effective in this case. Thus, we would like to verify the effect of our approach on different kinds of keywords. By using a sentence-level sentiment classifier [*7], we classify these keywords into three groups: *balanced*, *negative*, *positive*, where a positive keyword means that the sentences in the corpus have relatively more positive attitude towards it than others, etc. Then, we sample 20 the most frequent keywords from each group. However, there are much less negative response candidates in the corpus, so we end up with 13 keywords which have relatively more negative response candidates for the negative group. The distribution of attitude polarity of the response candidates is shown in **Fig. 4**. Note that even the negative group contains more positive response because positive sentiment dominates the corpus and the recall of negative polarity classification is much lower than positive polarity classification.

As a result, the test set consists of 53 keywords. For each keyword, 10 queries were sampled from the Xiaoice user log. We removed 54 queries that there is no attitude in their response candidates. Hence, the test set contains 53 keywords and 476 queries. These keywords include foods (11), countries/cities (8), compa-

[*6] $k = 80$ is an untuned parameter. We choose $k = 80$ because there are usually less than 80 relevant response candidates for each query in the corpus according to our experience in developing Xiaoice.

[*7] The performance of the sentence-level polarity classifier is 50.6% F1 tested on the 4,261 sentences of Xiaoice Index.

Table 3 F1 scores (%) of Attitude Detection and the subtasks Target Extraction (TE) and Polarity Classification (PC). Precision and recall are included in the parentheses. Scores are the mean values calculated from 10 random training runs. Note that some baseline models share the same trained models (BiLSTM, CNN, etc.) so they have the same evaluation scores on the subtasks. *** indicates AD-NET is statistically significantly better than the best among (b)–(f) at the significance level of 0.001 (the unpaired Tukey HSD test [36]).

			Attitude Detection			Target Extraction	Polarity Classification		
#	TE Model	PC Model	Overall F_1	F_1^+	F_1^-	F_1	Overall F_1	F_1^+	F_1^-
a	Human		48.4 (69.3, 38.0)	60.6 (75.5, 50.6)	36.3 (63.2, 25.4)	58.9 (66.3, 53.0)	57.7 (82.8, 45.3)	71.3 (88.8, 59.6)	44.1 (76.8, 30.9)
b	BiLSTM	CNN (37.2, 34.1)	34.1 (48.7, 45.5)	43.1 (25.6, 23.4)	25.2 (42.4, 51.3)	46.4 (58.7, 56.3)	56.1 (63.9, 76.1)	69.2 (53.6, 36.6)	43.1
c	BiLSTM-CRF	CNN (32.7, 32.3)	32.2 (41.6, 43.2)	42.2 (23.9, 21.5)	22.3 (61.3, 44.3)	51.3 (58.7, 56.3)	56.1 (63.9, 76.1)	69.2 (53.6, 36.6)	43.1
d	BiLSTM	BiLSTM-Attn (33.6, 32.4)	32.6 (41.6, 44.4)	42.7 (25.7, 20.3)	22.5 (42.4, 51.3)	46.4 (60.4, 53.5)	55.3 (66.7, 73.0)	68.9 (55.2, 34.0)	41.7
e	BiLSTM-CRF	BiLSTM-Attn (32.0, 28.4)	29.3 (35.6, 39.5)	37.3 (28.3, 17.3)	21.2 (61.3, 44.3)	51.3 (60.4, 53.5)	55.3 (66.7, 73.0)	68.9 (55.2, 34.0)	41.7
f	AD-NET-separate (Ptr-Net + CNN)		35.0 (41.4, 32.1)	45.0 (49.6, 44.0)	24.9 (33.3, 20.2)	52.6 (65.1, 45.8)	56.1 (58.7, 56.3)	69.2 (63.9, 76.1)	**43.1** (53.6, 36.6)
g	AD-NET		40.1*** (44.3***, 37.0)	50.8*** (52.9***, 48.8)	29.4*** (35.7, 25.2)	52.1 (57.7, 47.5)	**56.8** (62.8, 52.3)	**70.7** (73.8***, 68.0)	42.8 (51.8, 36.6)
h	AD-NET + syntactic features		**41.1** (47.8, 36.8)	**52.0** (56.2, 48.5)	**30.3** (39.5, 25.2)	**52.7** (59.5, 47.4)	56.0 (64.9, 50.3)	70.2 (75.8, 65.6)	41.8 (53.9, 35.0)

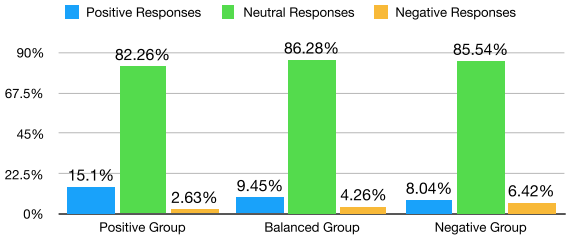

Fig. 4 Attitude polarity distribution of the response candidates in different keyword groups.

nies (5), zodiac signs (5), idols (6), sports (5), and others (13).

4.2.3 Evaluation

We directly compare the responses after re-ranking with the original responses from the retrieval module [*8] of Xiaoice using pairwise annotation. The evaluation consists of two parts: query-level and keyword-level. In query-level evaluation, each annotator is shown with a query and two responses at each time: one is the original response of Xiaoice and the other one is the response after re-ranking. The annotators are asked to choose the better responses according to the following criteria:

Coherence: A coherent response is logically connected and topically relevant to the original query [37].

Informativeness: An informative response provides new information in the eye of the user who issued the query [37].

Engagingness: An engaging response is interesting and results in an enjoyable conversation.

Consistency: We evaluate consistency in two granularities: (1) *query-level annotation consistency (Q-Consistency)*: for query, annotators are asked to judge if the corresponding response is consistent with the attitude we set in Attitude Profile. (2) *keyword-level evaluation (K-Consistency)*: responses under the same keyword will be annotated together, and annotators are asked to choose which group looks more consistent. For example, responding with "I don't like dogs" immediately after responding with "I like dogs" is not consistent [6].

Annotators do not know the responses are from which system since the responses are shuffled for each query and keyword. For both query-level and keyword-level annotations, "tie" can be chosen if there is no difference between the responses.

[*8] Xiaoice is a complex chatbot system, but we focus on only the essential part: retrieval module.

5. Results and Analysis

5.1 Attitude Detection

Table 3 shows the evaluation scores of Attitude Detection and its subtasks. The proposed model AD-NET achieved 40.1% on overall F1 and outperforms all the baselines systems and its separately trained variant AD-NET-separate. However, the separately trained model AD-NET-separate also achieved comparable scores in both TE and PC subtasks while it obtained lower overall F_1 in Attitude Detection. In the TE subtask, AD-NET-separate obtained 52.6% F1 score while AD-NET only achieved 52.1%. In the PC subtask, AD-NET-separate achieved 56.1% which is only marginally lower than AD-NET's 56.8%. However, the overall F_1 of AD-NET-separate in Attitude Detection is only 35.0%, which is considerably lower than that of AD-NET's 40.1% (the unpaired Tukey HSD test, $p < 0.001$). These results suggest that good performance in the PC and TE subtasks is not sufficient for obtaining high scores in the main task Attitude Detection.

We randomly sampled 25 conversations from the test collection and show the prediction error distributions for both AD-NET-separate and AD-NET models in **Fig. 5**. Here, a *Mismatched Error* means that the attitude prediction error consists of a correct subtask prediction and an incorrect one, and a *Matched Error* means that both subtasks predictions are incorrect. It can be observed that while both models have close numbers of errors in two subtasks, AD-NET made fewer attitude prediction errors (10) than AD-NET-separate (15) by reducing the number of mismatched errors. For the whole test collection, we observed that AD-NET made 256 mismatched errors out of 770 errors, but AD-NET-separate made 514 mismatched errors out of 916 errors. These results suggest that joint training may alleviate the target-

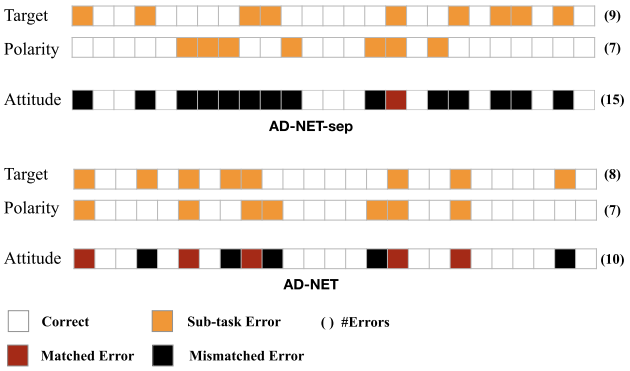

Fig. 5 Error type distributions of 25 randomly sampled conversations for AD-NET-separate and AD-NET models. Each column represents a conversation. A *Mismatched Error* means that the attitude prediction consists of a correct subtask prediction and an incorrect one. A *Matched Error* means that both subtasks predictions are incorrect.

Table 4 Inference and training time cost (unit: millisecond/conversation) for Attitude Detection.

	training	inference
BiLSTM + CNN	9.47	3.38
BiLSTM-CRF + CNN	11.52	3.71
BiLSTM-CRF + BiLSTM-Attn	10.07	3.53
AD-NET-separate (Ptr-Net + CNN)	7.93	3.09
AD-NET	**4.21**	**1.63**
AD-NET+ syntactic features	6.67	2.89

polarity mismatch problem: if the target is predicted correctly, the polarity will also tend to be correct. Thus, the joint-trained AD-NET model is more effective than the separated trained model when we evaluate polarity and target as a whole.

AD-NET + syntactic features scored 1.0% higher than AD-NET, but the improvement is not statistically significant according to the unpaired Tukey HSD test. It suggests that simply concatenating the syntactic features to the word embedding provides limited improvement on Attitude Detection.

5.1.1 Model Efficiency

Since applications related to sentiment analysis are usually applied to real-time large-scale data, the efficiency of model training and inference is also critical. The time costs of training and inference for each method are given in **Table 4**. It can be observed that the proposed model is consistently faster than the baseline systems. Due to sharing encoders by joint training, it is also faster than the AD-NET-separate model. In addition, AD-NET-separate (Ptr-Net + CNN) also works faster than BiLSTM based models, which suggests that Ptr-Net is indeed a less computationally expensive choice for target extraction.

5.1.2 Human Performance

The human performance shown in Table 3 is unexpectedly low. After inspecting the annotations, we believe that the main reason is that annotators have very different levels of sensitivity for sentiment. Some annotators try to catch the subtle sentiment expressed by the utterances. For example, in the development set, an annotator judged that the utterance "I eat beef noodles for breakfast everyday at home" expresses a positive attitude towards "beef noodles". However, the other two disagreed as the utterance does not provide explicit information about if he likes beef noodles and simply conveys a fact that he easts it everyday. For another example, in Fig. 3 (c), one annotator thought "fine" express positive attitude but the other two considered it neutral. The different interpretations may have caused low inter-rater agreement.

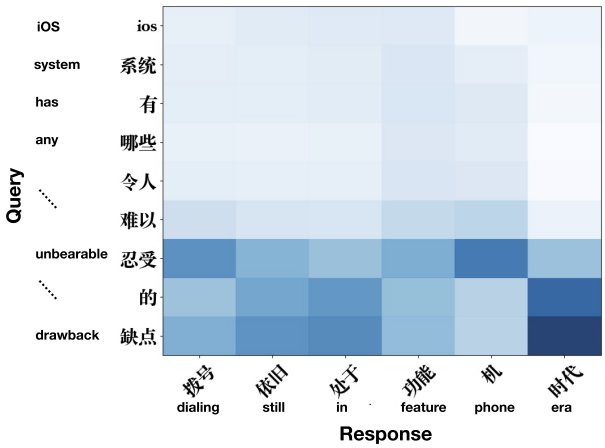

Fig. 6 Example of attention weights of the response encoder. Query: "*What unbearable drawbacks does iOS system have?*" Response: "*The dialing function is still in the feature phone era*". A darker colour means a higher attention weight.

In addition, some of the explicit targets are difficult to be extracted as simple text spans. For example, in Fig. 3 (d), two annotators considered "fried rice cake" the attitude target but the other annotator thought "the one my dad cooked" should be the target. Since the true target is "the fried rice cake my dad cooked" and cannot be extracted as a single span, the annotators picked some parts randomly, which may lead to disagreements.

5.1.3 Attention Visualisation

We visualise attention weights of response encoder with an example query-response pair in **Fig. 6**. The keywords "unbearable drawback" received higher weights during encoding the response. While Attitude Detection requires to predict the polarity of the response, words in the response do not express much negative information. With the key information of "unbearable drawback" which is obtained via attention mechanism, the response presentation now expresses a negative attitude in our proposed model.

5.2 Re-ranking with Attitudes

The results of comparing the re-ranked responses with the original responses are shown in **Table 5**. Firstly, we see that re-ranking with attitude indeed improves both K-Consistency and Q-Consistency as the re-ranking approach encourages the chatbot to express its attitudes about the keyword. Furthermore, coherence, informativeness and engagingness are also enhanced after re-ranking, which indicates that it is helpful to show the chatbot's attitude when the user mentioned the target in the chatbot's attitude profile. **Table 6** shows several examples that re-ranking improves/decreases the response quality. In the first example, the original response is relevant to the query, but not logically connected to the query. In contrast, the re-ranked response expresses a positive attitude to the entity mentioned in the query (i.e., badminton), which is also a natural way for human to respond. Furthermore, the original responses sometimes bring redundant information, like the "mocha" in the second example, but re-ranked responses may avoid this problem. Furthermore, when the original response outperforms attitude response, the margin may be

Table 5 Human evaluation results of re-ranking with attitudes. Win/Tie/Loss [*9] are the number of queries/keywords improved, unchanged, or hurt, compared to the responses before re-ranking according to the evaluation annotators. Rate of success% is calculated as 100% × WIN / (WIN+ LOSE). * and *** indicate statistically significant improvements according to two-sided binomial test at the significance level of 0.05 and 0.001, respectively.

	Win	Tie	Lose	Rate of success% with 95% CI
Coherence	210	167	99	68.0 [62.4, 73.1]***
Informativeness	198	123	155	56.1 [50.7, 61.3]*
Engagingness	204	112	160	56.0 [50.8, 61.2]*
Q-Consistency (All)	245	199	32	88.4 [84.1, 92.0]***
Q-Consistency (Balanced)	80	84	9	89.9 [81.7, 95.3]***
Q-Consistency (Negative)	72	35	9	88.9 [80.0, 94.8]***
Q-Consistency (Positive)	93	80	14	86.9 [79.0, 92.7]***
K-Consistency (All)	33	12	8	80.5 [65.1, 91.2]***
K-Consistency (Balanced)	13	2	5	72.2 [46.5, 90.3]
K-Consistency (Negative)	7	4	2	77.8 [40.0, 97.2]
K-Consistency (Positive)	13	6	1	92.9 [66.1, 99.8]*

Table 6 Examples of re-ranked responses v.s. original responses. win/lose indicates that the annotator judged that the re-ranked response is better/worse in all query-level criteria: coherence, informativeness and engagingness.

query	Do you like playing badminton?
original response	Do you often play badminton?
re-ranked response	I like badminton but I seldom play it. (win)
query	A cup of vanilla cafe latte, no coriander.
original response	mocha or vanilla cafe latte.
re-ranked response	I don't eat coriander. (win)
query	I want to travel to Yunnan.
original response	I am in Yunnan. You may visit us.
re-ranked response	Next goal: Yunnan. (lose)

small like the third example.

The balanced keyword group is as good as the other two groups in term of consistency, which is contrary to our expectation. After inspecting the conversations, we found that neutral utterances dominates the balanced group, which makes the proposed approach difficult to show good consistency to the annotators. Furthermore, while we call it "balanced" group, it is still contain more positive responses than negative responses, as Fig. 4 suggested. We hypothesise that even the baseline can easily achieve high consistency in the positive keyword group as there are many similar sentiment. However, there are still a lot of contradictory attitudes in the response candidates before re-ranking. In other words, there are still a lot of cases in the positive keyword group that need to be corrected by re-ranking. Thus, re-ranking outperformed the baseline significantly for the positive keyword group.

6. Conclusion

We tackled the problem of Attitude Detection, which we define as the task of extracting the replier's attitude, i.e., a target-polarity pair, from a given one-round conversation. While previous work considered Target Extraction and Polarity Classification separately, we regard them as subtasks of Attitude Detection. First, our experimental results show that treating the two subtasks independently is not the optimal solution for Attitude Detection, as achieving high performance in each subtask is not sufficient for obtaining high performance in Attitude Detection. Second, we proposed AD-NET which can be jointly trained for the two subtasks in an end-to-end manner. Experiments show that this model achieves a higher overall F1 score than models trained in isolation by alleviating the the target-polarity mismatch problem with joint training. In addition, by employing pointer networks to consider the target extracting task a boundary prediction problem, the model obtained better performance and faster training/inference than LSTM and LSTM-CRF based models.

We utilise the attitude detection model AD-NET to improve a retrieval-based chatbot by re-ranking the response candidates with the extracted attitude features. To verify this approach, we build an attitude profile manually, and use its attitude targets as keywords to sample queries from the user log of a commercial chatbot. Human evaluation indicates that the re-ranked responses to the sampled queries are statistically significantly more consistent, coherent, engaging and informative than the original ones obtained from a commercial chatbot.

As our approach to re-ranking response candidates are only verified with a retrieval-based system in the present study, we would like to further investigate whether it will work with generation-based chatbots in the future.

[*9] Note that unlike other metrics, Q-Consistency is not annotated in a paired manner, but Win of Q-Consistency means the proposed method responded with a correct attitude but the baseline didn't, and Tie of Q-Consistency denotes that both methods responded with a correct attitude or incorrect attitude.

References

[1] Shang, L., Lu, Z. and Li, H.: Neural Responding Machine for Short-Text Conversation, *ACL 2015* (2015).

[2] Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J. and Jurafsky, D.: Deep reinforcement learning for dialogue generation, arXiv preprint arXiv:1606.01541 (2016).

[3] Ke, P., Guan, J., Huang, M. and Zhu, X.: Generating Informative Responses with Controlled Sentence Function, *ACL 2018* (2018).

[4] Li, J., Galley, M., Brockett, C., Spithourakis, G.P., Gao, J. and Dolan, B.: A persona-based neural conversation model, arXiv preprint arXiv:1603.06155 (2016).

[5] Qian, Q., Huang, M. and Zhu, X.: Assigning personality/identity to a chatting machine for coherent conversation generation, *IJCAI-ECAI 2018* (2018).

[6] Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D. and Weston, J.: Personalizing Dialogue Agents: I have a dog, do you have pets too?, arXiv preprint arXiv:1801.07243 (2018).

[7] Zeng, Z., Song, R., Lin, P. and Sakai, T.: Attitude Detection for One-Round Conversation: Jointly Extracting Target-Polarity Pairs, *WSDM 2019* (2019).

[8] Hu, M. and Liu, B.: Mining and summarizing customer reviews, *SIGKDD 2004*, pp.168–177 (2004).

[9] Li, C., Guo, X. and Mei, Q.: Deep Memory Networks for Attitude Identification, *WSDM 2017*, pp.671–680 (2017).

[10] Ding, X., Liu, B. and Yu, P.S.: A holistic lexicon-based approach to opinion mining, *WSDM 2008* (2008).

[11] Kiritchenko, S., Zhu, X., Cherry, C. and Mohammad, S.: NRC-Canada-2014: Detecting aspects and sentiment in customer reviews, *SemEval 2014* (2014).

[12] Zhang, M., Zhang, Y. and Vo, D.-T.: Gated Neural Networks for Targeted Sentiment Analysis, *AAAI 2016*, pp.3087–3093 (2016).

[13] Tang, D., Qin, B., Feng, X. and Liu, T.: Effective LSTMs for Target-Dependent Sentiment Classification, *COLING 2016* (2016).

[14] Tang, D., Qin, B. and Liu, T.: Aspect Level Sentiment Classification with Deep Memory Network, *EMNLP 2016* (2016).

[15] Chen, P., Sun, Z., Bing, L. and Yang, W.: Recurrent Attention Network on Memory for Aspect Sentiment Analysis, *EMNLP 2017* (2017).

[16] Jakob, N. and Gurevych, I.: Extracting opinion targets in a single-and cross-domain setting with conditional random fields, *EMNLP 2010*, pp.1035–1045 (2010).

[17] Toh, Z. and Wang, W.: Dlirec: Aspect term extraction and term polarity classification system, *SemEval 2014* (2014).

[18] Poria, S., Cambria, E. and Gelbukh, A.F.: Aspect extraction for opinion mining with a deep convolutional neural network, *Knowledge*

Based System, Vol.108 (2016).
[19] Wang, W., Pan, S.J., Dahlmeier, D. and Xiao, X.: Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis, *EMNLP 2016* (2016).
[20] Vanzo, A., Croce, D. and 0001, R.B.: A context-based model for Sentiment Analysis in Twitter, *COLING 2014* (2014).
[21] McDonald, R.T., Hannan, K., Neylon, T., Wells, M. and Reynar, J.C.: Structured Models for Fine-to-Coarse Sentiment Analysis, *ACL 2007* (2007).
[22] Yang, B. and Cardie, C.: Context-aware Learning for Sentence-level Sentiment Analysis with Posterior Regularization, *ACL 2014*, pp.325–335 (2014).
[23] Feng, S., Wang, Y., Liu, L., Wang, D. and Yu, G.: Attention based hierarchical LSTM network for context-aware microblog sentiment classification, *World Wide Web* (2018).
[24] Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kočiskỳ, T. and Blunsom, P.: Reasoning about entailment with neural attention, *ICLR 2016* (2016).
[25] Wang, S. and Jiang, J.: Machine Comprehension Using Match-LSTM and Answer Pointer, *ICLR 2017* (2017).
[26] Wang, W., Yang, N., Wei, F., Chang, B. and Zhou, M.: Gated Self-Matching Networks for Reading Comprehension and Question Answering, *ACL 2017*, pp.189–198 (2017).
[27] Vinyals, O., Fortunato, M. and Jaitly, N.: Pointer networks, *NIPS 2015*, pp.2692–2700 (2015).
[28] Kim, Y.: Convolutional Neural Networks for Sentence Classification, *EMNLP 2014*, pp.1746–1751 (2014).
[29] Che, W., Li, Z. and Liu, T.: LTP: A Chinese Language Technology Platform, *COLING 2010*, pp.13–16 (2010).
[30] Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I. and Manandhar, S.: SemEval-2014 Task 4: Aspect Based Sentiment Analysis, *SemEval 2014* (2014).
[31] Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S. and Androutsopoulos, I.: Semeval-2015 Task 12: Aspect based sentiment analysis, *SemEval 2015* (2015).
[32] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Mohammad, A.-S., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., et al.: SemEval-2016 Task 5: Aspect based sentiment analysis, *SemEval 2016* (2016).
[33] Baziotis, C., Pelekis, N. and Doulkeridis, C.: DataStories at SemEval-2017 Task 4 - Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis, *SemEval 2017*, pp.747–754 (2017).
[34] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J.: Distributed representations of words and phrases and their compositionality, *NIPS 2013* (2013).
[35] Kingma, D. and Ba, J.: Adam: A method for stochastic optimization, *ICLR 2015* (2015).
[36] Sakai, T.: *Laboratory Experiments in Information Retrieval*, Vol.40, Springer (2018).
[37] Shang, L., Sakai, T., Lu, Z., Li, H., Higashinaka, R. and Miyao, Y.: Overview of the NTCIR-12 Short Text Conversation Task, *NTCIR-12* (2016).

Zhaohao Zeng is a second-year Ph.D. student at the Department of Computer Science and Engineering, Waseda University.

Ruihua Song is chief scientist of Microsoft XiaoIce. Before joining XiaoIce, she worked for Microsoft Research Asia from 2003 to 2017 as a lead researcher. In May 2017, her works on image inspired poetry generation was used to generate the first AI created and published collection of poems, "The Sunshine Lost Windows". Her research interests include information retrieval, data mining and artificial intelligence, especially AI based creation and multimodality. She served international conferences, such as CIKM (as area chair), SIGIR (as a senior PC), WWW, KDD, AAAI, etc., and journals, such as Information Retrieval (as editorial board), TOIS, TKDE, etc. She has published more than 50 papers.

Pingping Lin is a software developer of Microsoft XiaoIce.

Tetsuya Sakai Tetsuya Sakai is a professor and the head of department at the Department of Computer Science and Engineering, Waseda University, Japan. He is also a visiting professor at the National Institute of Informatics. He is an editor-in-chief of Springer's Information Retrieval Journal. He has received several research awards, including an IPSJ SIG Research Award (2006), IPSJ Best Paper Awards (2006, 2007), and a FIT Funai Best Paper Award (2008). He is an ACM Distinguished Member, and the ACM SIGIR Vice Chair (2019–2022).