Responsive and Self-Expressive Dialogue Generation

Kozo Chikai
SHIMANO INC.
tokoroten0401@gmail.com

Junya Takayama and **Yuki Arase**⋆
Osaka University
⋆Artificial Intelligence Research Center (AIRC), AIST
{takayama.junya,arase}@ist.osaka-u.ac.jp

Abstract

Generic responses frequently generated by neural models are a critical problem for user engagement in dialogue systems. For a more engaging chit-chat experience, we propose a response generation model motivated by the interpersonal process model for intimacy. It generates responsive and self-expressive replies, which are implemented as domain-awareness and sentiment-richness, respectively. Experiments empirically confirmed that our model outperformed the sequence-to-sequence model; 68.1% of our responses were domain-aware with sentiment polarities, which was only 2.7% for responses generated by the sequence-to-sequence model.

Domain: Baseball

User: Sugano will be the ace for Japan!
（菅野は日本のエースになるよ）

Seq2Seq: Really? （そうなの？）
Our model (+): Sugano is cool!
（菅野は良いですよね！）
Our model (-): No way? There is no hope for Sugano!
（菅野は無理ですね！）

Domain: Pokémon GO

User: Caught my first Pikachu! So cute!!
（初めてピカチュウ捕まえた！可愛い！！）

Seq2Seq: Wow, awesome!（え 、まじでいいよ !）
Our model (+): Shiny Pikachu is pretty cute.
（ピカチュウの色違いがなかなか可愛いですよね）
Our model (-): Shiny Pikachu is pretty hard, indeed.
（ピカチュウの色違いが、なかなか難しいですよね）

Figure 1: Responses generated by our model and SEQ2SEQ ((+) represents a positive response and (-) represents a negative response.)

1 Introduction

Dialogue systems that conduct non-goal-oriented chat, *i.e.*, chit-chat, is an active research area. The sequence-to-sequence model (SEQ2SEQ) (Vinyals and Le, 2015; Shang et al., 2015) is commonly used for implementation, however, recent studies, *e.g.*, (Li et al., 2016a), point out that SEQ2SEQ frequently generates overly generic responses. Among different approaches to address this problem, previous studies propose to generate more engaging responses by reacting to topics in users' utterances (Xing et al., 2017) or embodying emotions (Zhou et al., 2018; Huang et al., 2018). Herein we make a step further to generate responsive and self-expressive replies simultaneously.[1]

The interpersonal process model for intimacy (Reis and Shaver, 1988) indicates that *conversational responsiveness* (Miller and Berg, 1984), *i.e.*, showing concern for what was said, and *self-expression*, *i.e.*, sharing thoughts and feelings, are primary factors to create intimacy. Motivated by this theory, we believe that the conversational responsiveness and self-expression are also valid for a dialogue system to generate engaging responses. We implement the conversational responsiveness as *domain-awareness* because it effectively conveys an impression that the dialogue agent is listening to the user by responding about mentioned topics. Also, we implement the self-expression as *sentiment-richness* by representing sentiment polarity to generate subjective responses with feelings.

Specifically, the encoder predicts the domain of a user's utterance and integrates domain and utterance representations to tell the decoder the target domain explicitly. Then the decoder embodies sentiment polarity in its generation process. Fig. 1 shows real responses generated by our model. You may find that our responses react to the domains of input utterances while showing salient sentiments. On the other hand, SEQ2SEQ ends up generating generic responses.

To the best of our knowledge, this is the first study that simultaneously achieved both domain-

[1] In this study, we focus on single-turn conversations, *i.e.*, generating a response to a single utterance from the user.

Proceedings of the 1st Workshop on NLP for Conversational AI, pages 139–149
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

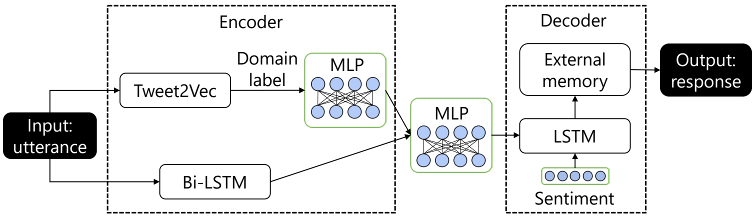


Figure 2: Architecture of the proposed model, on which the encoder is responsible for domain-awareness and the decoder takes care of embodying sentiment polarity.

aware and sentiment-rich response generation. Our contributions are twofold. First, we achieve these features in a simple architecture integrating existing methods on top of SEQ2SEQ in order to make it easily reproducible in existing dialogue systems. Second, our model utilizes fine-tuning to compensate for the training data scarcity, which is essential because there is a limited amount of domain-dependent and sentiment-rich dialogues. Our codes and scripts are publicly available.[2]

Evaluation results empirically confirmed that our model significantly outperformed SEQ2SEQ from the human perspective. Annotators judged that responses generated by our model are consistent with the utterances' domains and show salient sentiments for 89% and 72% of cases while preserving fluency and consistency. Furthermore, they judged 68.1% responses by our model as *both* domain-aware and sentiment-rich, which was only 2.7% for responses by SEQ2SEQ.

2 Related Work

The generic response problem in SEQ2SEQ is a central concern in recent studies. Different approaches have been proposed to generate diversified responses; by an objective function (Li et al., 2016a; Zhang et al., 2018b), segment-level reranking via a stochastic beam-search in a decoder (Shao et al., 2017), or by incorporating auto-encoders so that latent vectors are expressive enough for the utterance and response (Zou et al., 2018). In these approaches, balancing the diversity and coherency in a response is not trivial. Zou et al. (2018) show that metrics to measure the diversity are not proportional to human evaluation.

Another group of studies tackles the generic response problem by improving coherence in the response, which is relevant to conversational responsiveness. Approaches include reinforcement learning (Zhang et al., 2018a) and prediction of a keyword that will be the gist of a response given an input utterance and its generation in the decoder (Mou et al., 2016; Yao et al., 2017; Wang et al., 2018). In our study, we consider domain-level coherency to achieve the conversational responsiveness similar to (Xing et al., 2017).

Several studies focus on self-expression in responses. Some add persona in dialogue agents to generate consistent responses to paraphrased input utterances (Li et al., 2016b; Zhang et al., 2018c; Qian et al., 2018). Zhou et al. (2018) conducted the first study that controls emotions in dialogue agents using two factors. The first is embedding of a desired emotion label as in (Li et al., 2016b; Huang et al., 2018). The second is internal and external memories, which control the emotional state and the output of the decoder, respectively. These previous studies propose methods to achieve *either* conversational responsiveness or self-expression. Herein we aim to achieve *both* features simultaneously.

3 Proposed Architecture

To be easily implemented on existing dialogue systems, our model design aims to be simple. We integrate TWEET2VEC (Dhingra et al., 2016) and the external memory (Zhou et al., 2018) with SEQ2SEQ (Fig. 2). While sentiments in texts are well-understood in natural language processing, emotions need more studies to be considered in practical applications. Besides, determining the appropriate emotions for a specific utterance remains problematic (Hasegawa et al., 2013). In our model, we focus on sentiments and input the embedding of a sentiment label s to the decoder, which specifies the desired sentiment to represent in a response.

3.1 Encoder

Fig. 3 shows the design of our encoder, which integrates the input utterance and its domain.

[2] https://github.com/KChikai/Responsive-Dialogue-Generation

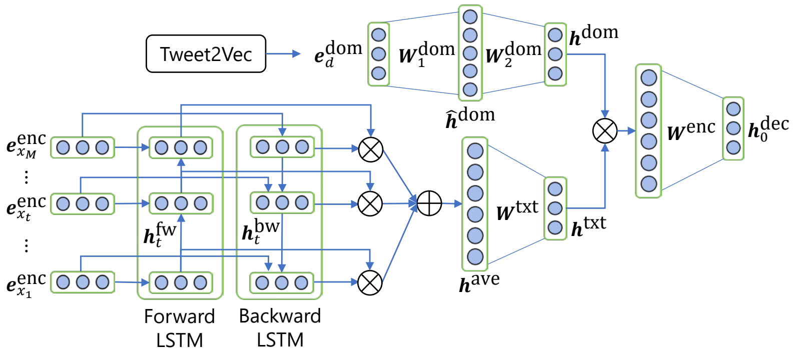


Figure 3: Design of the encoder ($\otimes$ concatenates input vectors and $\oplus$ averages them)

Input Utterance Encoding The input utterance is represented as a vector. Bi-directional recurrent neural networks empirically show superior performance in generation tasks (Bahdanau et al., 2015) because they refer to the preceding and subsequent sequences. We apply bi-directional Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks to encode an input utterance into a vector. Given the input utterance $X = \{x_1, x_2, \cdots, x_M\}$ of length M, the forward LSTM network encodes the input at time step t as

$$\mathbf{c}_t^{\text{fw}}, \mathbf{h}_t^{\text{fw}} = \text{LSTM}(\mathbf{e}_{x_t}^{\text{enc}}, \mathbf{c}_{t-1}^{\text{fw}}, \mathbf{h}_{t-1}^{\text{fw}}).$$

$\mathbf{h}_t^{\text{fw}} \in \mathbb{R}^{\lambda}$ is the representation output, which is computed based on the embedding of x_t (denoted as $\mathbf{e}_{x_t}^{\text{enc}} \in \mathbb{R}^{\omega}$) and the previous representation output $\mathbf{h}_{t-1}^{\text{fw}}$. $\mathbf{c}_{t-1}^{\text{fw}} \in \mathbb{R}^{\lambda}$ is a cell state vector that works as a memory in LSTM. The backward LSTM works in the same fashion by reading the input in the reverse order. The final vector representation $\mathbf{h}^{\text{txt}} \in \mathbb{R}^{\lambda}$ is computed by averaging the concatenated forward and backward outputs

$$\begin{aligned}\mathbf{h}^{\text{ave}} =& \frac{1}{M}\sum_{t=1}^{M}[\mathbf{h}_t^{\text{fw}}; \mathbf{h}_t^{\text{bw}}],\\ \mathbf{h}^{\text{txt}} =& \sigma(\mathbf{W}^{\text{txt}}\mathbf{h}^{\text{ave}}),\end{aligned}$$

where $[\cdot;\cdot]$ concatenates two vectors, $\sigma(\cdot)$ is a sigmoid function, and $\mathbf{W}^{\text{txt}} \in \mathbb{R}^{\lambda\times 2\lambda}$. In this way, $\mathbf{h}^{\text{txt}}$ encodes the summaries of both the preceding and subsequent words.

Domain Estimation & Representation Another task of the encoder is predicting the domain of the input utterance and integrating the domain label with the utterance. For domain estimation, we apply TWEET2VEC due to its superior ability to predict a label of short and colloquial text, which should be the case for input utterance to dialogue agents. Although the original paper predicted hashtags of tweets, we predict domains of utterances. Another advantage of TWEET2VEC is that it is language-independent and easily adapted to different languages.

Specifically, TWEET2VEC encodes the input utterance using bi-directional recurrent neural networks adapting gated recurrent units (GRUs) (Cho et al., 2014). The final vector representation of input $\hat{\mathbf{h}}^{\text{txt}}$ is computed by integrating the forward and backward outputs using a fully-connected layer. Then $\hat{\mathbf{h}}^{\text{txt}}$ is passed through a linear layer, and the posterior probabilities of the domains are computed in a softmax layer.

Domain d of the highest posterior probability is converted into dense vector representation $\mathbf{h}^{\text{dom}} \in \mathbb{R}^{\delta}$. Specifically, a two-layer multilayer perceptron (MLP) is employed where a rectifier is used as the activation function

$$\begin{aligned}\hat{\mathbf{h}}^{\text{dom}} =& \text{relu}(\mathbf{W}_1^{\text{dom}}\mathbf{e}_d^{\text{dom}}),\\ \mathbf{h}^{\text{dom}} =& \text{relu}(\mathbf{W}_2^{\text{dom}}\hat{\mathbf{h}}^{\text{dom}}),\end{aligned}$$

where $\mathbf{e}_d^{\text{dom}} \in \mathbb{R}^{\delta}$ is the embedding vector of d, $\mathbf{W}_1^{\text{dom}} \in \mathbb{R}^{\eta\times\delta}$, and $\mathbf{W}_2^{\text{dom}} \in \mathbb{R}^{\delta\times\eta}$.

Utterance & Domain-Label Integration Finally, the utterance and domain representations pass through another fully-connected layer and are integrated into a vector $\mathbf{h}_0^{\text{dec}} \in \mathbb{R}^{\lambda}$

$$\mathbf{h}_0^{\text{dec}} = \mathbf{W}^{\text{enc}}[\mathbf{h}^{\text{txt}}; \mathbf{h}^{\text{dom}}], \tag{1}$$

where $\mathbf{W}^{\text{enc}} \in \mathbb{R}^{\lambda\times(\lambda+\delta)}$. $\mathbf{h}_0^{\text{dec}}$ is then passed to the decoder for response generation.

3.2 Decoder

Given $\mathbf{h}_0^{\text{dec}}$ encodes the input utterance and the predicted domain, the decoder generates a response embodying the desired sentiment. Input utterance X is paired with a sequence of outputs to predict $Y = \{y_1, y_2, \ldots, y_N\}$ of length N.

We apply the external memory to SEQ2SEQ in order to proactively control the sentiments in the outputs. Fig. 4 shows the detailed design of the decoder. First, we concatenate the output sequence with the embedding of the desired sentiment label as a soft-constraint to instruct the decoder of the desired sentiment for response generation (Li et al., 2016b). The external memory then directly controls response generation by switching outputs between words with sentiment polarities (hereafter referred to as *sentiment words*) and generic ones. Specifically, in the external memory, vocabulary V is divided into two subsets: $V = \{V_s \cup V_g\}$. V_s contains only sentiment words, such as `cool` and `terrible`, while V_g contains other generic words, such as `day` and `me`. The weight of a switcher, which determines the priority of the sets of vocabulary is computed based on the representation output from an LSTM network.

Embedding of s (denoted as $\mathbf{e}^s \in \mathbb{R}^\delta$) is concatenated with output y_{t-1} at the previous time step and then input into the LSTM network as

$$\mathbf{c}_t^{\text{dec}}, \mathbf{h}_t^{\text{dec}} = \text{LSTM}([\mathbf{e}_{y_{t-1}}^{\text{dec}}; \mathbf{e}^s], \mathbf{c}_{t-1}^{\text{dec}}, \mathbf{h}_{t-1}^{\text{dec}}),$$

where $\mathbf{c}_t^{\text{dec}} \in \mathbb{R}^\lambda$ is the cell state vector in the LSTM, $\mathbf{h}_t^{\text{dec}} \in \mathbb{R}^\lambda$ is the representation output from the LSTM, and $\mathbf{e}_{y_{t-1}}^{\text{dec}}$ is the embedding of y_{t-1}. Recall that the initial input to the decoder $\mathbf{h}_0^{\text{dec}}$ is computed in Eq. (1).

Then $\mathbf{h}_t^{\text{dec}}$ is passed to the external memory to sequentially predict output as

$$\begin{aligned} a_t =& \sigma(\mathbf{W}^a \mathbf{h}_t^{\text{dec}}), \\ \mathbf{o}_g =& \text{softmax}(\mathbf{W}^g \mathbf{h}_t^{\text{dec}}), \\ \mathbf{o}_s =& \text{softmax}(\mathbf{W}^s \mathbf{h}_t^{\text{dec}}), \\ y_t \sim \mathbf{o_t} =& [(1 - a_t)\mathbf{o}_g; a_t \mathbf{o}_s], \end{aligned}$$

where $\mathbf{W}^a \in \mathbb{R}^{1\times\lambda}$, $\mathbf{W}^g \in \mathbb{R}^{\lambda\times|V_g|}$, and $\mathbf{W}^s \in \mathbb{R}^{\lambda\times|V_s|}$. $a_t \in [0, 1]$ weighs either the probabilities of generic words or sentiment words based on context represented in $\mathbf{h}_t^{\text{dec}}$. $\mathbf{o}_g \in \mathbb{R}^{|V_g|}$ and $\mathbf{o}_s \in \mathbb{R}^{|V_s|}$ are the posterior probabilities to output a word in each vocabulary. $\mathbf{o}_t \in \mathbb{R}^{|V|}$ is the final probability of each word adjusted by a_t. At run-time, a beam-search with a beam-size of 5 is conducted to avoid outputting an unknown tag.

Our model optimizes the cross-entropy loss between predicted word distribution $\mathbf{o}_t$ and gold distribution $\mathbf{p}_t$. In addition, a regularizer constrains the selection of a sentiment or generic word

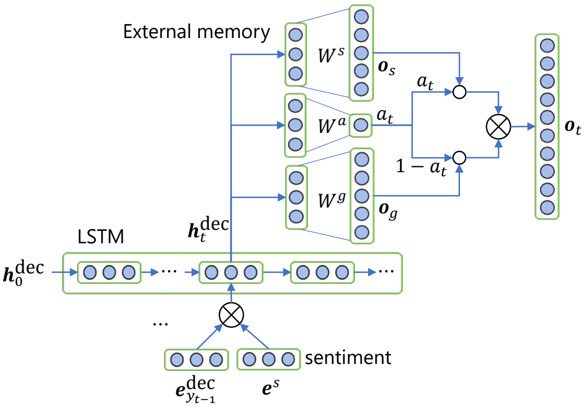


Figure 4: Design of the decoder (⊗ concatenates input vectors and ∘ multiplies a vector and scalar.)

$$-\sum_{t=1}^{N} \mathbf{p}_t \log(\mathbf{o}_t) - \sum_{t=1}^{N} q_t \log(a_t), \qquad (2)$$

where $q_t \in \{0, 1\}$ is the gold choice of a sentiment word or a generic word.

4 Training Framework

Because our model aims to generate domain-aware responses with sentiments, it should be trained on in-domain conversations with sentiments. Although either in-domain conversations or conversations with sentiments are available, their intersections are scarce. Furthermore, our model integrates TWEET2VEC and external memory. Thus, training errors propagate from each sub-model to the final response.

Consequently, we designed a training framework that pre-trains sub-models independently and then conducts fine-tuning on the connected model, where a model is trained using the pre-trained parameters as the initial weights. The training process uses not only a small-scale conversational (in-domain) corpus of specific domains but also a large-scale conversational corpus of general domain.

4.1 Sentiment Annotation

Training requires sentiment annotations on the general and in-domain corpora. Because it is cost prohibitive to annotate sentiments to these corpora manually, we rely on automatic sentiment analysis. Given that input utterances to dialogue agents are short, incomplete, extremely casual, and potentially noisy, we need a robust method to predict sentiments with guaranteed accuracy.

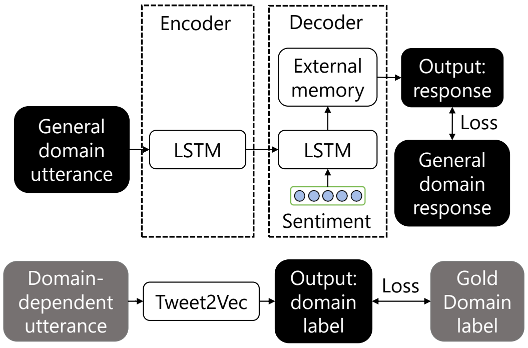


Figure 5: Pre-training process (Gray boxes denote data from the in-domain corpus.)

Although we tried several state-of-the-art methods for sentiment analysis (Severyn and Moschitti, 2015; Zhu et al., 2015), our preliminary evaluation showed that they were easily confused by colloquial styles in conversational texts. Hence, we used a simple heuristics based on a sentiment lexicon to prioritize the robustness in analysis. Specifically, a sentence is annotated as positive (negative) if there are more positive (negative) words. If there is an equal number of positive and negative words, then the sentence is annotated as neutral.

We extracted words with strong polarities from existing sentiment lexicons (Kobayashi et al., 2005; Takamura et al., 2005). Besides, we collect casual and recent sentiment words by crawling Twitter.[3] This sentiment lexicon is used for the above sentiment analysis and the external memory as V_s after the filtering described in Sec. 5.2. More details of lexicon construction are in Sec. A.

4.2 Pre-Training on Sub-Models

After annotating sentiments on the general and in-domain corpora, we conducted pre-training. In the pre-training step, sub-models are independently trained (Fig. 5).

Seq2Seq requires large-scale training data for fluent response generation. Thus, we used the general corpus here. We directly connected the bi-directional LSTM in the encoder and the LSTM in the decoder to train this sub-model. The loss function (Eq. (2)) is computed by referring to the gold-responses in the corpus. Embeddings to represent sentiments are trained at this stage.

Tweet2Vec is independently trained using the in-domain corpus for domain prediction. The model optimizes the categorical cross-entropy loss between the predicted and gold domain labels.

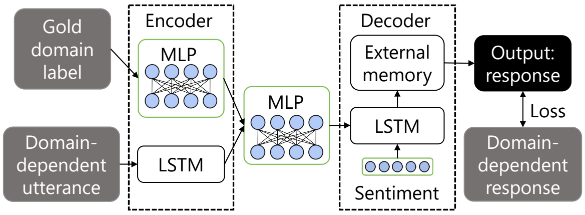


Figure 6: Fine-tuning process (Gray boxes denote data from the in-domain corpus.)

4.3 Fine-Tuning on the Entire Model

After pre-training, fine-tuning is conducted using the in-domain corpus to train MLPs that integrate the domain label and input utterance (Fig. 6). Additionally, embeddings of domain labels $\mathbf{e}_d^{\text{dom}}$ are trained at this stage. To avoid error propagation from the pre-trained Tweet2Vec, gold domain labels are inputted into the MLP to learn correct representations of domain labels.

Once fine-tuned, these sub-models are connected to generate domain-aware responses with sentiments (Fig. 2).

5 Evaluation Design

Because the effectiveness of each component for embodying emotions have been evaluated in (Zhou et al., 2018; Huang et al., 2018), we focus on evaluating whether *both* domain-awareness and sentiment-richness are achieved *simultaneously* by our model compared to Seq2Seq.

5.1 Data Collection

To train our model, we collected both general and in-domain conversational texts in Japanese. The general corpus is constructed by crawling conversational tweets using Twitter API.[4] We also crawled conversational tweets used in the NTCIR Short Text Conversation Task (Shang et al., 2016). In total, the general corpus contains about 1.6M utterance-response pairs.

The in-domain corpus crawls conversations in public Facebook Groups using Facebook Graph API.[5] Because members are fans of specific products, organizations, and people, we expect that their conversations are domain-dependent.[6] Specifically, we used two domains, Japanese pro-

[3] https://twitter.com/

[4] https://developer.twitter.com/en/docs

[5] https://developers.facebook.com/docs/graph-api

[6] We also tried to collect in-domain conversations using hashtags on Twitter, but they were too noisy.

Type	Domain	# of Pairs
General	Mixture	1.1M
In-domain	Baseball	24k
	Pokémon Go	23k

Table 1: Training data profile

	Summary	Setting
ω	Dimension of word embedding	256
λ	Dimension of the representation output in the LSTM network	512
δ	Dimension of the embedding and representation output of labels	64
η	Dimension of the hidden layer in the MLPs	512
$\|V\|$	Vocabulary size	45k
$\|V_s\|$	Vocabulary size of the sentiment words	$1,387$

Table 2: Hyper-parameters and their settings

fessional baseball leagues and Pokémon Go[7], anticipating that salient sentiments are easily manifested in sports and game domains. Experiments using a wider range of domains is our future work. We crawled conversations since a group's inception to November 2017. In total, the in-domain corpora contain about 29k baseball-related conversations and 28k game-related conversations. We assume that sentiments can be embodied in domains with weaker sentiment tendencies due to pre-training in the general domain corpus. Verification of this assumption is a future task.

After crawling, we preprocessed the corpora to remove noise and standardize texts (details are described in Sec. B). Table 1 shows the amount of our training data after the preprocessing step.

As a validation set of pre-training, 1k conversation pairs were sampled from the general corpus. Similarly, 1k pairs for validation and another 1k as a test set were sampled from the in-domain corpus for the automatic evaluation. The training set excluded these validation and test sets.

5.2 Model Setting

Table 2 summarizes the hyper-parameters in our model and their settings. The vocabulary size was 45k, which consisted of frequent words in the general and in-domain corpora. The general and in-domain corpora contained $1,387$ sentiment words, which were used as V_s in the external memory.

In both pre-training and fine-tuning, submodels, except for TWEET2VEC, were trained at most 100 epochs with early stopping using the validation set. Batch size was set to 200, dropout was used with a rate of 0.2, and Adam (Kingma and Ba, 2015) with a learning rate of 0.01 was applied as an optimizer.

During pre-training and fine-tuning, an out-of-vocabulary (OOV) word in input utterances was replaced with a similar word in the vocabulary to reduce the effects of data sparsity (Li et al., 2016c). We generated word embeddings using the fastText (Bojanowski et al., 2017) with the default settings feeding Wikipedia dumps[8] as training data. When a word is OOV, the top-50 similar words are detected using cosine similarities between their embeddings. If one of these similar words is in the vocabulary, it replaces the original OOV word. Otherwise, the original word is replaced with an unknown word tag.

TWEET2VEC was trained on the in-domain corpus using the official implementation[9] with the default settings. We crawled 200 new domain-dependent conversational pairs as a validation set. The prediction accuracy was 89.0%, which is reasonable considering that our texts are colloquial.

We compare our model to SEQ2SEQ that was implemented using bi-directional LSTM networks as an encoder and an LSTM network as a decoder. Our model has the same hyper-parameters and training procedures, except that SEQ2SEQ was trained using both general and in-domain corpora. For SEQ2SEQ, a validation set of 1k pairs was randomly sampled from the combined corpus excluding from the training and test sets described in Sec. 5.1.

5.3 Human Evaluation

Because each utterance has many appropriate responses, an automatic evaluation scheme has yet to be established. To assess the quality of the generated responses from the human perspective, we designed two evaluation tasks. Task 1 evaluates the overall quality of our model compared to SEQ2SEQ from the perspectives of

[7] https://pokemongolive.com/en/

[8] https://dumps.wikimedia.org/

[9] https://github.com/bdhingra/tweet2vec

domain-awareness and sentiment-richness. Task 2 evaluates if an intended sentiment is embodied as desired without being affected by domain-awareness.

We recruited five graduate students majoring in computer science that are Japanese native speakers (hereafter called annotators). After an instruction session to explain judgment standards, they annotated Task 1 and Task 2. As a token of appreciation, each annotator received a small stipend.

Test Set Creation To exclude external factors, *e.g.*, word segmentation failures, that may affect the evaluation results, we manually created a test set consisting of 300 utterances in the baseball domain and another 300 utterances in the Pokémon Go domain.

First, we crawled new conversational pairs from the same Facebook Groups from November to December 2017. Next, we manually excluded conversations in the general domain (*e.g.*, greetings). We then cleaned sentences in the same manner with the general and in-domain corpora. Besides, we manually replaced OOV words within vocabulary words that preserve the original meanings of sentences. Slang and uncommon expressions were also manually converted to standard expressions to avoid impacting the accuracy of word segmentation. Half of the test set (150 conversations for each domain) was used for Task 1 and the other half was used for Task 2. Note that all annotators annotated the same conversations, in total 600 pairs of utterances and responses.

Task 1: Overall Evaluation Annotators judged triples of an input utterance and responses by our model and by SEQ2SEQ. The order of responses was randomly shuffled to ensure a fair evaluation. Annotators assessed the following aspects:

- Fluency: Annotators judged if a response is fluent and at an acceptable level to understand its meaning (1 = fluent, 0 = influent).
- Consistency: Annotators evaluated whether a response is semantically consistent with the utterance (1 = consistent, 0 = inconsistent). Generic responses can be regarded as consistent if they are acceptable for given utterances. Responses judged as influent are automatically annotated as inconsistent.
- Domain-awareness: Annotators compared the two responses and determined which one better matched the domain of the input utterance (1 = model that generated the better response, 0 = the other model).
- Sentiment-richness: Annotators compared the two responses and determined one showing salient sentiments like Domain-awareness annotation. Only positive or negative responses were considered for our model.

For Domain-awareness and Sentiment-richness, we conduct a pairwise comparison of our model and SEQ2SEQ, which enables reliable judgments for subjective annotations (Ghazvininejad et al., 2018; Wang et al., 2018), rather than independently judging different models.

Task 2: Evaluation of Sentiment Control Our model takes a sentiment label that is desired to be expressed in a generated response as input, which we refer to as *intended* sentiment. This task evaluates if such an intended sentiment is embodied in a response by comparing the intended sentiment and a sentiment that annotators perceive in practice.

Annotators were shown a pair of input utterance and generated response by our model, and then asked to judge if the response was positive, negative, or neutral. We evaluated the agreement between the intended and perceived sentiments.

6 Evaluation Results

As an automatic evaluation measure, we computed the BLEU score (Papineni et al., 2002) following evaluations in (Li et al., 2016a; Ghazvininejad et al., 2018). Our model achieved the higher BLEU score (1.54) than SEQ2SEQ (1.39). However, as discussed in (Liu et al., 2016; Lowe et al., 2017), current automatic evaluation measures show either weak or no correlation with human judgements, or worse, they tend to favor generic responses. Hence, we focus on human evaluation in the following.

First of all, the agreement level of annotations is examined based on Fleiss' κ. All annotations have reasonable agreements ($\kappa \geq 0.37$) except the annotation of fluency for SEQ2SEQ whose κ value is as low as 0.21 (all the κ values are shown in Sec. C). This phenomenon may be because SEQ2SEQ tends to output generic responses that are less dependent on the utterances, making judgments difficult due to the limited clues to evaluate fluency.

Table 3 shows the macro-averages and the 95% confidence intervals of the scores obtained by the

Metrics	SEQ2SEQ	Our model
Fluency	**0.995** ± 0.006	0.955 ± 0.023
Consistency	**0.773** ± 0.094	0.753 ± 0.127
Domain-awareness	0.109 ± 0.044	**0.890** ± 0.044
Sentiment-richness	0.282 ± 0.133	**0.717** ± 0.133

Table 3: Evaluation results of Task 1

annotators in Task 1. Our model achieved significant improvements over SEQ2SEQ; 89% and 72% of the responses generated by ours were deemed as consistent with the utterance domain and showing salient sentiments, respectively. Furthermore, 68.1% responses by our model were judged as *both* domain-aware and sentiment-rich, which was only 2.7% for responses by SEQ2SEQ.

As for fluency and consistency, SEQ2SEQ yields slightly more fluent (99.5%) and consistent (77.3%) responses compared to our model (95.5% and 75.3%, respectively). SEQ2SEQ benefits from the generic responses because such responses apply to various inputs, making it easier to achieve a high consistency compared to our model that generates domain-dependent responses. Additionally, generic responses are easier to generate because they are typically short. The average numbers of characters in responses when inputting the test set were 19 and 32 for SEQ2SEQ and our model, respectively. This result reveals that our model achieves a reasonably high fluency even when generating significantly longer responses. Another reason is the side-effect of external memory that influences the internal state of the decoder as reported in (Zhou et al., 2018).

As a result of Task 2, the macro-average of the agreement between the intended and perceived sentiments is 64.5 ± 2.3%, where Fleiss' κ of annotation is 0.52. Fig. 7 is a confusion matrix showing the distribution of the obtained 1,500 annotations. Neutral responses tend to be judged as either positive (28.5%) or negative (15.6%). One reason is our simple sentiment annotation, which assigns a neutral label when the numbers of positive and negative words in a sentence are equal. Improving the polarity strength is a future task.

The annotators perceived 17.6% of the intended negative responses as positive. Detailed analyses of generated responses revealed that this category contained sentiment words whose polarities depend on the context, *e.g.*, `envy`, `great`, and `surprising`. These words are considered negative in our sentiment lexicon because they tend to be used with negative emoticons to show humor in Twitter. In the future, we will develop post-processing to clean our lexicon and consider the self-attention (Vaswani et al., 2017) to resolve such context-dependent cases.

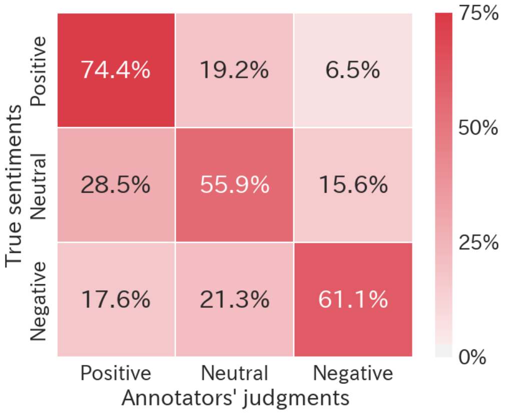


Figure 7: Confusion matrix of intended (true) sentiments and the sentiments that annotators perceived

Fig. 1 shows real examples of generated responses. While SEQ2SEQ produces generic responses like "`Really?`", our model generates domain-aware responses with sentiments like "`Sugano is cool!`" (positive response) and "`No way? There is no hope for Sugano!`" (negative response) for the baseball domain. Sec. D provides more examples that show how our model achieved domain-awareness and sentiment-richness.

7 Conclusion

As a solution to the generic response problem in SEQ2SEQ, we implemented conversational responsiveness and self-expression to a neural dialogue model. Different from previous studies, our model achieves these features simultaneously in forms of domain-awareness and sentiment-richness, respectively. Evaluation results empirically demonstrated that our model significantly outperformed SEQ2SEQ. In the future, we will improve the accuracy in embodying sentiments and extend our dataset to cover diverse domains.

Acknowledgments

This project is funded by Microsoft Research Asia, Microsoft Japan Co., Ltd., and JSPS KAKENHI Grant Number JP18K11435.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proc. of the Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST)*, pages 103–111.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016. Tweet2Vec: Character-based distributed representations for social media. In *Proc. of ACL*, pages 269–274.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Scott Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proc. of AAAI*.

Takayuki Hasegawa, Nobuhiro Kaji, Naoki Yoshinaga, and Masashi Toyoda. 2013. Predicting and eliciting addressee's emotion in online dialogue. In *Proc. of ACL*, pages 964–972.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Chenyang Huang, Osmar Zaiane, Amine Trabelsi, and Nouha Dziri. 2018. Automatic dialogue generation with expressed emotions. In *Proc. of NAACL-HLT*, pages 49–54. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Nozomi Kobayashi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi, and Toshikazu Fukushima. 2005. Collecting evaluative expressions for opinion extraction. In *Proc. of IJCNLP*, pages 596–605.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In *Proc. of EMNLP*, pages 230–237.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proc. of NAACL-HLT*, pages 110–119.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proc. of ACL*, pages 994–1003.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2016c. Towards zero unknown word in neural machine translation. In *Proc. of IJCAI*, pages 2852–2858.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proc. of EMNLP*, pages 2122–2132.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proc. of ACL*, pages 1116–1126.

Lynn C. Miller and John H. Berg. 1984. Selectivity and urgency in interpersonal exchange. In Valerian J. Derlega, editor, *Communication, Intimacy, and Close Relationships*, chapter 7, pages 161–205. Elsevier.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *Proc. of COLING*, pages 3349–3358.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Assigning personality/profile to a chatting machine for coherent conversation generation. In *Proc. of IJCAI*, pages 4279–4285.

Harry T. Reis and Phillip Shaver. 1988. Intimacy as an interpersonal process. In S. Duck, D. F. Hay, S. E. Hobfoll, W. Ickes, and B. M. Montgomery, editors, *Handbook of personal relationships: Theory, research and interventions*, pages 367–389. John Wiley & Sons.

Aliaksei Severyn and Alessandro Moschitti. 2015. UNITN: Training deep convolutional neural network for twitter sentiment classification. In *Proc. of the Int'l Workshop on Semantic Evaluation (SemEval)*, pages 464–469.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proc. of ACL-IJCNLP*, pages 1577–1586.

Lifeng Shang, Tetsuya Sakai, Zhengdong Lu, Hang Li, Ryuichiro Higashinaka, and Yusuke Miyao. 2016. Overview of the NTCIR-12 short text conversation task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, pages 473–484.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proc. of EMNLP*, pages 2210–2219.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proc. of ACL*, pages 133–140.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NIPS*, pages 5998–6008.

Oriol Vinyals and Quoc V Le. 2015. A neural conversational model. In *ICML Deep Learning Workshop*.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proc. of ACL*, pages 2193–2203.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proc. of AAAI*, pages 33351–33357.

Lili Yao, Yaoyuan Zhang, Yansong Feng, Dongyan Zhao, and Rui Yan. 2017. Towards implicit content-introducing for generative short-text conversation systems. In *Proc. of EMNLP*, pages 2190–2199.

Hainan Zhang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2018a. Reinforcing coherence for sequence to sequence model in dialogue generation. In *Proc. of IJCAI*, pages 4567–4573.

Hainan Zhang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2018b. Tailored sequence to sequence models to different conversation scenarios. In *Proc. of ACL*, pages 1479–1488.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018c. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proc. of ACL*, pages 2204–2213.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proc. of AAAI*.

Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *Proc. of ICML*, pages 1604–1612.

Meng Zou, Xihan Li, Haokun Liu, and Zhihong Deng. 2018. Memd: A diversity-promoting learning framework for short-text conversation. In *Proc. of COLING*, pages 1281–1291.

A Construction of the Sentiment Lexicon

We used two sentiment lexicons created by Kobayashi et al. (2005) and Takamura et al. (2005). The former is manually created, while the latter is automatically created by estimating the strengths of semantic orientations of words in the range of $[-1.0, 1.0]$. We only used words with a strong polarity. Specifically, words with scores of $[-1.0, 0.9]$ or $[0.9, 1.0]$. These lexicons contain only formal words like headings in dictionaries. Therefore, we extended our sentiment lexicon to collect casual and recent sentiment words.

We searched tweets that are expected to contain sentiments by querying Twitter with positive and negative emoticons. In total, we crawled 400k potential positive and negative tweets and generated word embeddings from these tweets using the fastText (Bojanowski et al., 2017) with the default setting. We then manually selected 57 sentiment words from the vocabulary as seeds. The top-15 similar words per seed were extracted as sentiment words, which were ranked by the cosine similarity between embeddings of the seed and a candidate. In total, we collected $1,621$ negative and $2,666$ positive words as our sentiment lexicon.

B Preprocessing

We employed conversational text crawled from Twitter and Facebook, which are inherently noisy. We conducted data cleaning before training our model.

First, line breaks, emoticons, Japanese emoticons (kaomoji), URLs, and consecutive duplicate symbols were removed. Then texts less than or equal to 25 words were obtained after word segmentation using Mecab (Kudo et al., 2004). Table 4 shows detailed statistics of our training data after this preprocessing.

C Annotation Agreement

Table 5 shows the Fleiss' κ for each annotation result in our human evaluation. It confirms that reasonably high agreements were achieved.

D Example Responses

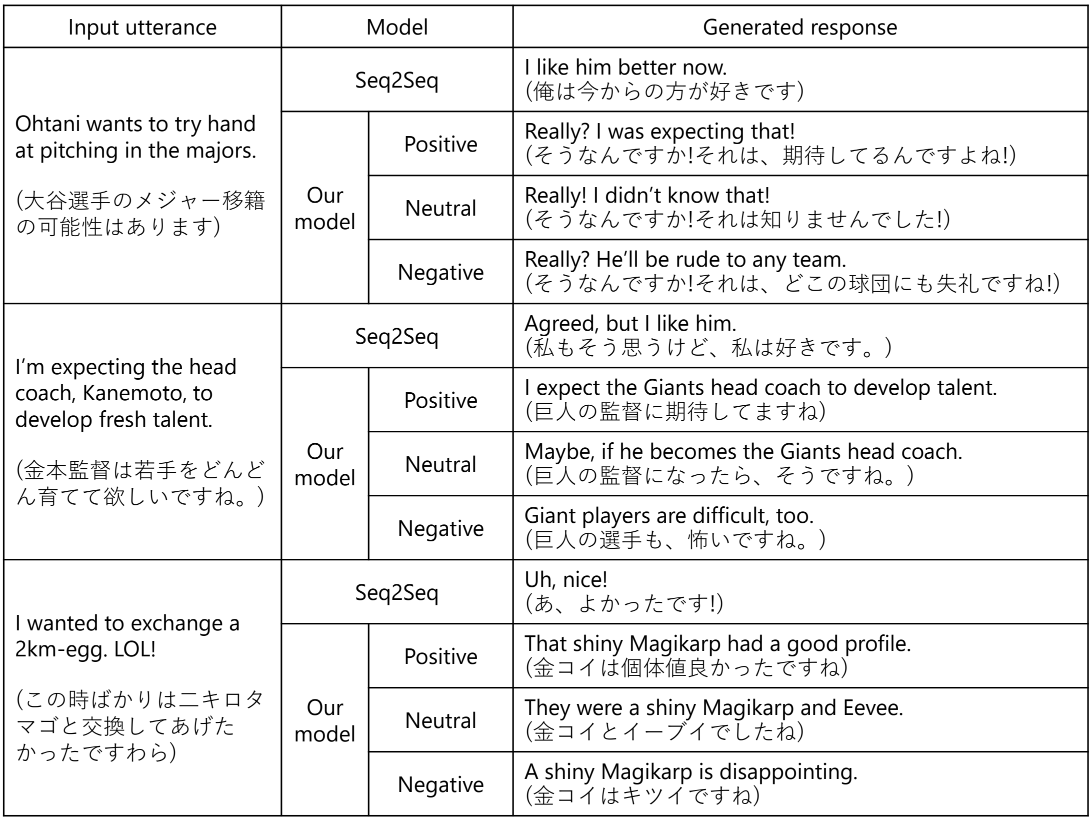
Fig. 8 shows real responses generated by our model and SEQ2SEQ using input utterances from our test set. Our method successfully generates

Data (Source)		Total # of pairs	Positive	Neutral	Negative
General (Twitter)		$1,081,500$	$218,322$	$760,451$	$102,727$
In-domain (Facebook)	Baseball	$24,666$	$6,011$	$16,623$	$2,032$
	Pokémon GO	$22,734$	$7,633$	$13,299$	$1,802$

Table 4: Detailed statistics of training data

Annotation			κ
Task 1	Fluency	SEQ2SEQ	0.21
		Our model	0.45
	Consistency	SEQ2SEQ	0.41
		Our model	0.37
	Domain-awareness		0.78
	Sentiment		0.46
Task 2			0.52

Table 5: Fleiss' κ on annotation results



Input utterance	Model		Generated response
Ohtani wants to try hand at pitching in the majors. (大谷選手のメジャー移籍の可能性はあります)	Seq2Seq		I like him better now. (俺は今からの方が好きです)
	Our model	Positive	Really? I was expecting that! (そうなんですか!それは、期待してるんですよね!)
		Neutral	Really! I didn't know that! (そうなんですか!それは知りませんでした!)
		Negative	Really? He'll be rude to any team. (そうなんですか!それは、どこの球団にも失礼ですね!)
I'm expecting the head coach, Kanemoto, to develop fresh talent. (金本監督は若手をどんどん育てて欲しいですね。)	Seq2Seq		Agreed, but I like him. (私もそう思うけど、私は好きです。)
	Our model	Positive	I expect the Giants head coach to develop talent. (巨人の監督に期待してますね)
		Neutral	Maybe, if he becomes the Giants head coach. (巨人の監督になったら、そうですね。)
		Negative	Giant players are difficult, too. (巨人の選手も、怖いですね。)
I wanted to exchange a 2km-egg. LOL! (この時ばかりは二キロタマゴと交換してあげたかったですわら)	Seq2Seq		Uh, nice! (あ、よかったです!)
	Our model	Positive	That shiny Magikarp had a good profile. (金コイは個体値良かったですね)
		Neutral	They were a shiny Magikarp and Eevee. (金コイとイーブイでしたね)
		Negative	A shiny Magikarp is disappointing. (金コイはキツイですね)

Figure 8: Example responses generated by our model and SEQ2SEQ

domain-aware and sentiment-rich responses compared to SEQ2SEQ.