Journal Pre-proof

A Survey on Empathetic Dialogue Systems

Yukun Ma, Khanh Linh Nguyen, Frank Z. Xing, Erik Cambria

PII: S1566-2535(20)30309-2
DOI: https://doi.org/10.1016/j.inffus.2020.06.011
Reference: INFFUS 1251

To appear in: *Information Fusion*

Received date: 30 November 2019
Revised date: 20 May 2020
Accepted date: 23 June 2020

Please cite this article as: Yukun Ma, Khanh Linh Nguyen, Frank Z. Xing, Erik Cambria, A Survey on Empathetic Dialogue Systems, *Information Fusion* (2020), doi: https://doi.org/10.1016/j.inffus.2020.06.011

Highlights

1) Identified 3 key features of empathetic dialog systems.

2) Discussed emotion-, personality-awareness and knowledge-accessibility as sub-topics

3) Technical summaries for recent progresses on conversational AI research

4) Provided tables to organize studies into chronological order

A Survey on Empathetic Dialogue Systems

Yukun Ma[†], Khanh Linh Nguyen[†], Frank Z. Xing[‡], Erik Cambria[‡*]

†AIR Labs, Continental A.G.

‡School of Computer Science and Engineering, Nanyang Technological University

**Corresponding author*

Abstract

Dialogue systems have achieved growing success in many areas thanks to the rapid advances of machine learning techniques. In the quest for generating more human-like conversations, one of the major challenges is to learn to generate responses in a more empathetic manner. In this review article, we focus on the literature of empathetic dialogue systems, whose goal is to enhance the perception and expression of emotional states, personal preference, and knowledge. Accordingly, we identify three key features that underpin such systems: emotion-awareness, personality-awareness, and knowledge-accessibility. The main goal of this review is to serve as a comprehensive guide to research and development on empathetic dialogue systems and to suggest future directions in this domain.

Keywords: Artificial Intelligence, Affective Computing, Dialogue Systems.

1. Introduction

The primary goal of building a dialogue system is to address users' questions and concerns via emulating the way humans communicate with each other. As human language is too complicated to be considered as a single target, dialogue systems have to model different aspects of human communication separately. Recent years have witnessed the emergence of empathy models in the context of dialogue systems and, hence, an increasing attention from the natural language processing (NLP) community.

Empathy is the capability of projecting feelings and ideas of the other party to someone's knowledge [1]. It plays an important part in the communication of human beings as it has the potential for enhancing their emotional bond. As noted by a previous study [2], incorporating empathy into the design of a dialogue system is also vital for improving user experience in human-computer interaction. More importantly, being empathetic is a necessary step for the dialogue agent to be perceived as a social character by users [3]. Building an empathetic dialogue system is then premised on the idea that it will result in improved user engagement and, consequently, more effective communication. Research on dialogue system has elaborated on the concept on dialogue system mainly from perspective of features. For example, Loojie et al. [4] stated that an empathetic dialogue system should be complimentary, attentive, and compassionate. In this survey, we are particularly concerned with the unique dimension of dialogue systems from the perspective of functions. Namely, what function has enabled empathetic behavior of a dialogue system. To our knowledge, this has not been discussed in depth by previous literature.

Early attempts to build dialogue systems can be dated back to the 1960s [5]. Since then, dialogue systems are either designed to perform specific tasks such as flight booking [6], healthcare [7], political debate [8], hence termed "task-specific dialogue systems", or to chitchat as a way of entertainment [9], hence called "chatbots". A task-specific dialogue system [10, 11] often consists of multiple modules including language understanding, dialogue state tracking, dialogue policy, and dialogue generation. On the other hand, recent progress in deep learning [12] also facilitates the use of end-to-end solutions to dialogue systems which can be more easily trained to simulate the behavior of human communication via access to a large amount of training data. As we will discuss in later sections, the process of generating responses conditioned on the existing contexts of a dialogue can be naturally modeled as a translation process where off-the-shelf end-to-end solutions such as the sequence-to-sequence (Seq2Seq) model [13] have already been proven effective.

The rapid growth of dialogue systems and their applications have intrigued many comprehensive surveys in the past decade. Chen et al. [14] mainly organize their survey by elaborating on each functional component of a dialogue system. Gao et al. [15] proposed the most recent review with good coverage of related topics, mainly focused on neural network-based approaches for building dialogue systems. Unlike [14] and [15], we position our perspective on dialogue systems with empathetic features. Related work [16] viewed empathy to be equivalent to emotion. We argue that empathy is not all about emotions. Indeed, a non-empathetic dialogue system may disappoint and bore the user for that the responses are too robotic yet incoherent, and consequently leads to the loss of affection.

Introducing emotion into the generation of dialogue could only partially address the problem. As illustrated by Fig. 1, a more comprehensive empathetic framework also has to access general knowledge as well as personalized knowledge. Personalization, in such a case, could increase the coherence and consistency of a dialogue system. With knowledge of user-specific information, the dialogue system could tailor responses towards the user's preference and address questions relevant to the user's untold background, and a virtuous cycle comes into form when the user tends to provide more information and clue about themselves. Moreover, external knowledge, being it task-specific or commonsense, usually complements the contexts of a conversation with additional background. Many facts that are obvious to human beings may be very opaque to a machine, for example: "I come to my friend's house. Jimmy is my friend" will be understood as it is when it comes to vanilla dialogue systems. It will not conclude that "my friend's house" means "Jimmy's house" unless we construct a relationship between them. This is where the knowledge part comes into play: it helps dialogue systems become smarter, sharper, and more interesting. Although it seems prevalent to incorporate knowledge into dialogue systems, reasoning, retrieving and representing a large scale knowledge base remain challenging. All three components (i.e., emotion, personalization, and knowledge) work together to ensure a smooth and natural flow of the conversation.

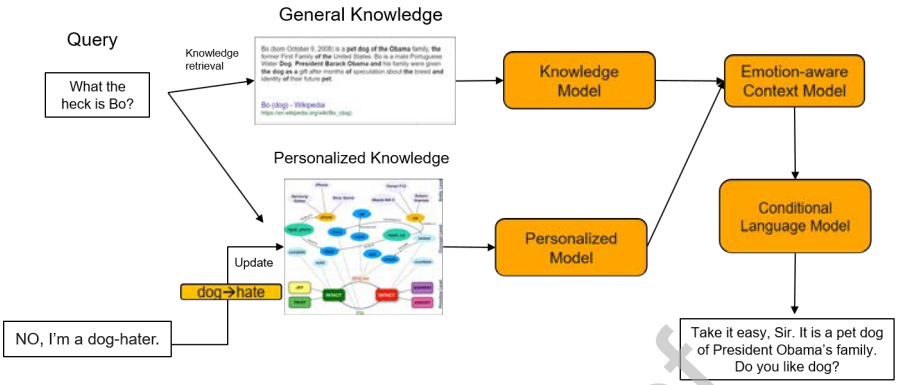


Figure 1: Typical workflow of an empathetic dialogue system.

Considering such complexity of empathetic systems, we take a perspective that goes beyond the merely emotional definition of empathetic dialogue systems by identifying three pillars. Such pillars accordingly represent the three main sub-topics presented in this survey:

- perceiving and expressing emotion (Section 3 – Affective Dialogue Systems)
- caring each individual (Section 4 – Personalized Dialogue Systems), and
- casting into knowledge (Section 5 – Knowledgeable Dialogue Systems).

In addition to previous surveys [14], we also cover the most recent advances in the area of empathetic dialogue systems. Especially, we would like to emphasize the end-to-end model more than traditional pipeline models as we believe the former represents the current trend of this field. To the best of our knowledge, we are the first to survey the empathetic features of a dialogue system. Overall, we primarily surveyed 35 papers selected from those published on prestigious venues in the past 10 years.

2. Propaedeutic Background

A dialogue system is not a system built on top of one model. Instead, it is built on integrating multiple techniques due to the complexity of language and tasks. In this section, we present a technical introduction to recent techniques that serve as the backbone of an empathetic dialogue system.

2.1. Neural Language Model

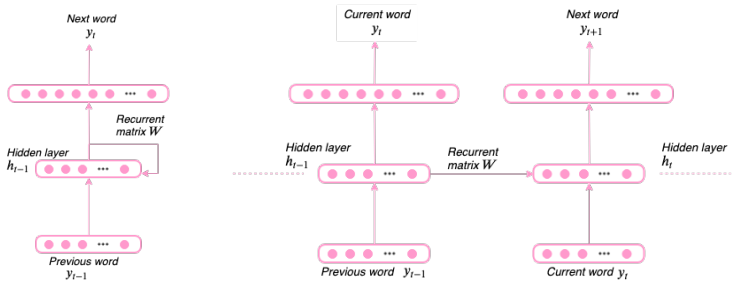
The language model generally defines a probability distribution over the sequence of words and thus plays an important part in a dialogue system. Tradition methods include N-gram models is based on statistics. Most recently, language models based on neural networks have achieved state-of-the-art performance in a variety of tasks.

2.1.1. Recurrent Neural Network

Perhaps, one of the most well-known language models is the recurrent neural network (RNN) language model [17] (Fig. 2). As a notable feature of RNN, the history (or contexts) of a word sequence is encoded into a hidden layer via an input transformation and a recurrent connection. Each word in a sentence is first mapped to the word embedding space and then updates the history vector. The history vector is then linearly transformed and normalized to represent a probability distribution of ejecting the next word. In theory, the greatest elegance of RNN is that it could encode contexts of arbitrary length. At each time-step t, RNN transforms the input vector and history vector as below:

$$H_t = \sigma(W_h * y_t + W * H_{t-1})$$
$$P(y_{t+1}) = softmax(W_o * H_t)$$

with h_t being the hidden layer at time-step t, W is the fixed for different time-steps, y_t is the word vector of current (input) word, and y_{t+1} represents the word vector of next (output) word. However, the linear transformation applied on the history vector is multiplied over time which leads to the gradient to vanish or explode when the sequence is long.



(a) Compact (b) Unrolled

Figure 2: The two views of RNN language model.

To overcome this problem, RNN variants such as gated recurrent unit (GRU) [18] and long short-term memory (LSTM) [19] have been proposed. A gating function is usually implemented as a sigmoid function that restricts the scale of gradients so that it would explode after multiple time steps. In LSTM, there are three gates: input gate, forget gate and output gate. At each time-step t, a LSTM cell transforms the input and memory cell as follows:

$$
\begin{aligned}
i_t &= \sigma(U_i * y_t + h_{t-1} * W_i) \\
f_t &= \sigma(U_f * y_t + h_{t-1} * W_f) \\
o_t &= \sigma(U_o * y_t + h_{t-1} * W_o) \\
\bar{c}_t &= \sigma(U_c * y_t + h_{t-1} * W_c) \\
h_t &= o_t * \tanh(c_t) \\
c_t &= f_t * c_{t-1} + i_t * \bar{c}_t
\end{aligned}
$$

where W_i, W_f, W_o, W_c are the linear transformation for hidden input, hidden forget gate, hidden output and hidden candidate gate, respectively. Similarly, U_i, U_f, U_o, U_c are the weight vectors for input gate, forget gate, output gate and candidate gate. y_t is the input vector, h_t is the current cell output, and c_t is the current cell memory. Similarly, h_{t-1} is the previous cell output, and c_{t-1} is the previous cell memory. The forget gate may choose to forget certain content of the memory cell and input gate control how information flows from current input to the memory cell.

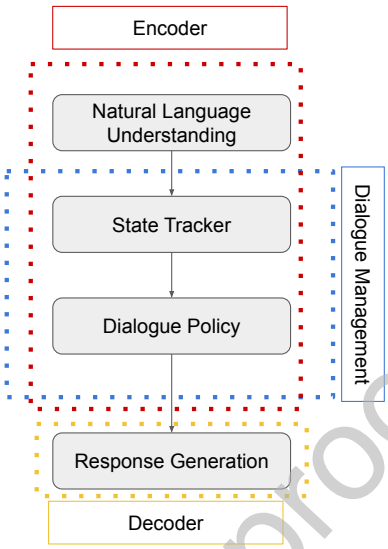


Figure 3: A unified view of the architecture of a dialogue system.

2.1.2. Sequence-to-Sequence Model

The generation of responses is conditioned on a given context. The probability distribution of generating a response can be seen as a conditional language model. Before going further into the details of language model, we would like to discuss the similarity and distinction between well-known encoder-decoder perspective of the dialogue system and the traditional modularized framework. Fig. 3 shows an unified view of the system's architecture. A modularized system usually have four parts: an natural language understanding (NLU) module that extracts structural information from the input; a dialogue state tracker (DST) that infers the dialogue states; a dialogue policy (DP) module that decides actions to be taken by the system; and a response generator to generate responses based on the output of all the precedent modules. The DST and DP together might be also referred to as the dialogue management module. From an encoder-decoder perspective, the system is consisting of an encoder and decoder where the encoder plays an equivalent role as the combination of NLU, DST and DP.

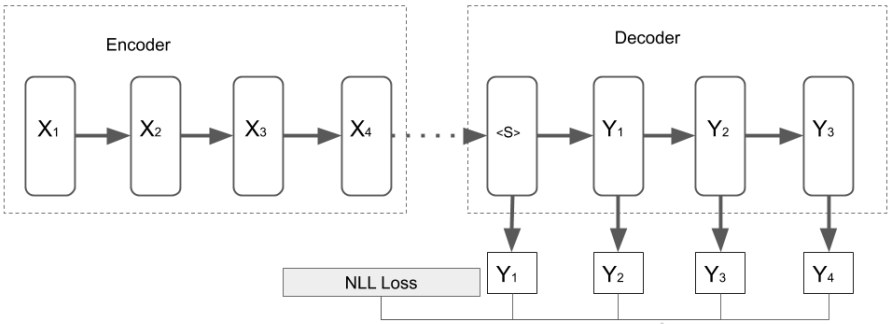


Figure 4: The architecture of Seq2Seq model.

The Seq2Seq model [13], sometimes also referred to as the encoder-decoder model, is probably the most widely used neural architecture for modeling the conditional dialogue generation. Although previous surveys have elaborated on Seq2Seq models, we still would like to provide a brief account of it for being self-content. Seq2Seq was initially proposed for sequence generation tasks such as machine translation. In the context of dialogue systems, the encoder of Seq2Seq encodes the context and emits words in the response. Fig. 4 shows the basic architecture of a Seq2Seq model. Given a dialogue context $X = x_1, x_2, \ldots, x_N$ consisting of N words, the model outputs a responsive sequence $Y = y_1, y_2, \ldots, y_M$ of length M. The model is composed of an encoder and a decoder, both of which are typically based on RNNs (including GRU and LSTM). The encoder converts X into a sequence of hidden outputs $H = h_1, h_2, \ldots, h_N$ where each $h_t = \phi(x_t, h_{t-1})$. The decoder then generates a sequence of hidden outputs $C = c_1, c_2, \ldots, c_M$ conditioned on H. At each time step, the decoder draws a word W_t from the distribution over vocabulary, which is calculated based on o_t,

$$y_t = \arg\max p(y_t|X, Y_{-t})$$

where $p(y_t|X, Y_{-t}) = softmax(o_t, Y_{t-1})$ and $o_t = f(o_t, H)$.

2.2. Attention Mechanism

The most straightforward way to represent a context of dialogue in a Seq2Seq model is to pass the last hidden output of the encoder to the decoder either by

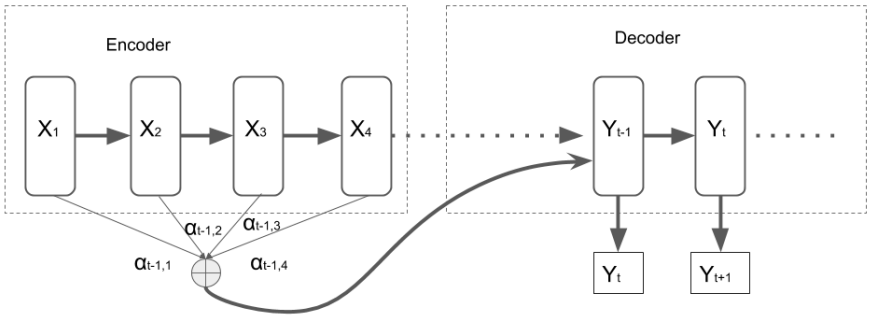


Figure 5: The architecture of attention mechanism.

concatenating with the input embedding or by using it as the initial hidden state of the decoder. However, the last hidden output h_N might be insufficient to encode information of the whole input sequence, especially when the length of the input sequence is long. Despite that gating function might help ease the problem, the recent study has suggested that the maximum number of words encoded by a LSTM network is still very limited. On the other hand, the emission of a word in the response may depend on a relevant excerpt of the context. One of the most effective solutions is to have an 'alignment' model [20] that allows the decoder to access a 'most relevant' position of the context. This can be achieved by computing a probability distribution over the encoder outputs denoting the probability of paying attention to one particular position, which is called attention mechanism. Both the soft version and stochastic version is possible to be used.

As shown in Fig. 5, at each time step t of decoding, the attention weight on the jth input word is computed as

$$\alpha_{tj} = \frac{exp(g(h_j, o_{t-1}))}{\sum_k exp(g(h_k, o_{t-1}))}$$

Note that *alpha* is normalized over the input sequence so that it sums up to one. Therefore, it could also be interpreted as a distribution over the positions of the input sequence.

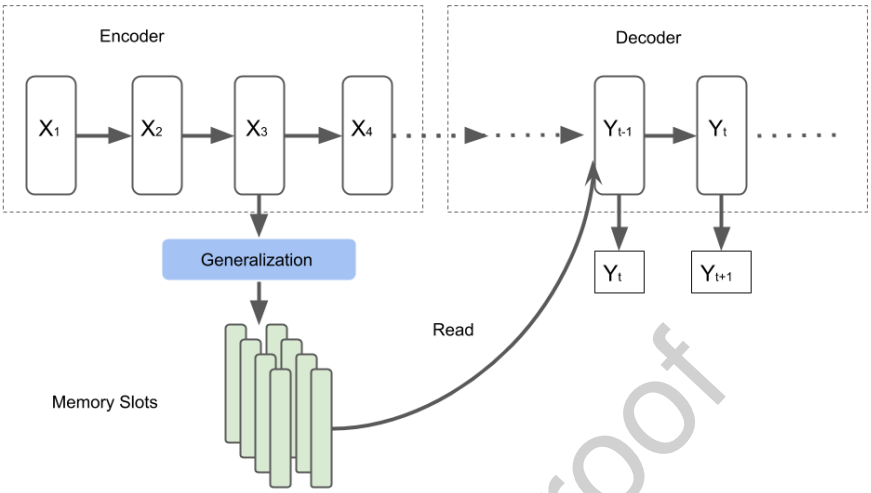


Figure 6: The architecture of the memory network model.

2.3. Memory Networks

One of the more general views of the hidden space of an RNN is that it is a memory to be updated over time. However, as we have pointed out, such memory might be too small and desegregate to store the necessary contents [21, 22] and thus may fail in application areas such as dialogue where a long term memory is required to understand the context. To address this problem, Weston et al. [22] proposed an architecture called memory network (MMN) which utilizes external memory slots and can be updated or read via writing a reading pointers.

As shown in Fig. 6, the input sequence is encoded and passed to the generalization module of MMN which controls the update (write and read) of memory slots. There is another read action performed by the decoding process which decides what content of memory to be loaded to predict the next word. The reading and writing actions can be simply implemented as an attention mechanism over the memory slots.

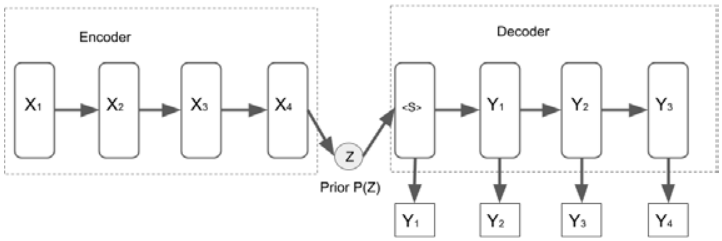


Figure 7: The architecture of variational autoencoder.

2.4. Variation Autoencoder

As shown in Fig. 7, variational autoencoder (VAE) [23] defines a conditional probability distribution $P(\boldsymbol{z}|X)$ which can be used for drawing a vector of latent codes $\boldsymbol{z}$ to represent the internal state of the dialogue context. One popular choice for $P(\boldsymbol{z}|X)$ is normal distribution $\mathcal{N}(\mu, \Sigma)$, where μ and Σ are the mean vector and covariance matrix, respectively. It is notable that presuming the forms of $P(\boldsymbol{z}|X)$ might be unreasonable, and it is also unnecessary. An auxiliary variable $\hat{\boldsymbol{z}}$ can be drawn from a simple standard normal distribution $\mathcal{N}(0, I)$, and an distribution of $\boldsymbol{z}$ of any form can then be obtained by a transformation function, $f(\hat{z})$. Moreover, all the parameters of $f(\cdot)$ can be learned in back-propagation. Note that normal distribution is continuous and not a natural choice for discrete variables. It works when the input data X is, for example, speech or image, while having difficulty in handling discrete data such as natural language. Most recently, related work has employed a re-parameterization technique called Gumbel-softmax [24] to allow back-propagation on the non-differentiable sampling processing of categorical distribution.

Besides the encoding process, VAE typically also comes with a generator. As it suggests, the generator aims at reconstructing the input X given the latent code $\boldsymbol{z}$. This autoencoding process enforces the sampling of $\boldsymbol{z}$ to encode information sufficient to produce X. The entire framework is trained by the Evidence Lower Bound Optimization (ELBO) on data log-likelihood,

$$E_{\boldsymbol{z}\sim q(\boldsymbol{z}|X)}[\log p(X|\boldsymbol{z})] - KL(q(\boldsymbol{z}|X)||p(\boldsymbol{z})).$$

Maximizing the above evidence lower bound is equivalent to maximizing the log-likelihood of generating X given the latent code $\boldsymbol{z}$ while, at the same time, minimizing the KL divergence between $q(\boldsymbol{z}|X)$ and the prior distribution $p(\boldsymbol{z})$. In the problem setting of building a dialogue system, what we need is a model that can take as input the dialogue history (including previous turns within a conversation) and generates a proper response. As such, VAE has to be extended to conditional variational autoencoder (CVAE) [25], which models the conditional probability distribution $P(Y|X)$ with X being the input and Y being the output response. The evidence lower bound of VAE can be rewritten as

$$E_{\boldsymbol{z}\sim q(\boldsymbol{z}|X,Y)}[\log p(Y|\boldsymbol{z},X)] - KL(q(\boldsymbol{z}|X,Y)||p(\boldsymbol{z}|X)).$$

During training, the recognition model $q(\boldsymbol{z}|X,Y)$ of CVAE is trained to approach the prior model $p(\boldsymbol{z}|X)$, while during testing, the generated response Y is not available as input, so the latent code outputted by the prior model will be passed to the generator. Let us take a close look at the objective function of ELBO, it should be noted that the maximization of log-likelihood is usually implemented by factoring $\log p(Y|z,X)$ into an element-wise form $\sum_i \log p(y_i|z,X)$. It, of course, suffers from the problem that the element-wise loss fails to have a holistic view of the generation error.

2.5. Generative Adversarial Network

As illustrated in Fig. 8, the architecture of a generative adversarial network (GAN) [26] is composed of a generator G and a discriminator D. It has achieved great success in multiple tasks ranging from image generation to transfer learning. The whole architecture is optimized based on a min-max game where the generator (G) is trained to fool the discriminator (D) by maximizing the classification error while D is trained to minimize the classification errors, defined as

$$\min_{\theta_G}\max_{\theta_D} E_{X\sim p(X)}[\log D(X|\theta_D)] + E_{z\sim p(z)}[\log(1 - D(G(z|\theta_G)|\theta_D)$$

The generator G defines a generation function $G(z,\theta_G)$ parameterized by θ_G. G draws samples from a probability distribution p_g based on a noise z. In the

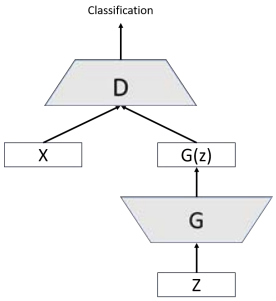


Figure 8: Generative adversarial network.

next phase, generated samples are fed to the discriminator $D(y, \theta_D)$, while D is trained to classify the generated samples as 'false' and real data and 'true'. The discriminator also utilizes the samples drawn from real-life distribution p_data to improve its discriminative ability in telling if the sample is drawn from p_data. Intuitively, the generator receives feedback from the discriminator on how well the generated sample can confuse the discriminator.

On the one hand, the discriminator of GAN takes as input the generated sentence as a whole and measures its closeness to responses generated by human beings. On the other hand, the GAN framework is equivalent to optimizing the Jensen-Shannon divergence of which the global optimality of G if achieved when $p_g = p_{data}$ [26].

2.6. Reinforcement Learning

Dialogue generation models based on Seq2Seq and its extensions (e.g., VAE) have faced one fundamental issue on the learning objective. Commonly used objective functions including likelihood and ELBO do not have a clear link with the realistic goal of a dialogue system. A direct result is that the model trained using such objective functions are usually those frequently seen in the training data [27]. For example, "i'm not sure" may be a response with high likelihood but is not a good response in the sense that it would not engage the user in further talking. Besides the dialogue generation model, the inherent states of user might also be introduced to dialogue management module of

some systems [28, 29] which defines dialogue actions to be taken by the system. Reinforcement learning, which has emerged as a powerful tool for both flexible reward function into the learning process and effectively modeling the transition of actions, has achieved enormous success in a variety of fields including dialogue system [30, 27]. For dialogue systems, reinforcement learning defines an 'environment' in which the dialogue agent can explore and receive feedbacks interactively. Since the key element of a dialogue is interaction, reinforcement learning could facilitate the use of ideal interactive patterns as rewards in learning the actions to be taken by the dialogue agent. A dialogue agent is supposed to take a "dialogue action" at each time step which results in a new state. The outcome of changing to a new state is measured by a reward function and is given to the agent as feedback from the environment. The objective is then selecting actions that could maximize the cumulative future rewards. In the context of building a dialogue system, a dialogue action a is usually generating a responsive utterance. In such sense, the action space is infinite for that the generated utterance is of arbitrary length and word choices. The state s can be any feature representation of the current conversation. For example, the dialogue state can be represented as the previous dialogue turns or previously extracted entities or slot values. When the action-to-take is concerned, a policy may be implemented as a probability distribution of taking an action a_t given the current state s_t, i.e., $p(a_t|s_t)$. In order to learn a proper action model, a variety of rewards have been explored in the literature for guiding the generation of meaningful and coherent responses, which we will cover in more detail in later sections.

However, one thing to note is that the action of generating an utterance is usually decomposed to a sequence of micro-actions that generate one word at a time. It is reasonable to assume that the environment is mostly not able to provide timely feedback to the generation of each word or even each utterance. For example, in a conversation about flight booking, the agent will only receive the reward after the whole conversation is completed. In such a case, a delayed reward function might be used to update the action model.

3. Affective Dialogue System

Emotion plays an important role in cognition and social behavior [31]. Existing study suggest that emotion is a reaction and a social and cultural interaction that is continuously developing by the relationships between human and the surrounding environment [32]. Yet, the definition and categorization of emotions remain fuzzy and long-debated among psychologists and philosophers [33]. In the scope of this paper, we focus on the representation of emotion in dialogue system (or human-computer interaction) and its effectiveness, not emotion in terms of affective science or social sciences in general [34].

Moreover, emotion is argued to have more social functions such as eliciting people's particular response [35], coercing actions or recruiting social support [31]. Most importantly, existing study suggests that emotion might be a related measure of decision making [36]. These theories support that incorporating emotions is advantageous to dialogue system by allowing the dialogue system to emulate the conversational behavior of human beings and, at the same time, to strengthen the emotional connection with human users [37]. Emotion might also increase the user's engagement in the conversation [38, 39]. On the other hand, it has been argued that emotion might introduce unpredictability into the system. Therefore, it has be to be placed in control with careful system design [40]. In this survey, we intend to follow the literature by referring to such a dialogue system, which is capable of perceiving, understanding, expressing and regulating emotion, as an affective dialogue system [29]. We then further define two groups of core features of an affective dialogue system. The first type of features, called emotion-awareness, is concerned with the representation of emotion expressed in the context of the conversation. Namely, the dialogue system should be able to detect the user's current emotional states given the conversation. The second type of features, called emotion-expressiveness, are mainly concerned with incorporating emotional information into generated responses. In fact, early studies have attempted to do so, either by handling users' emotions [41, 42] or infusing emotional-rich features into the virtual agents [43, 44].

Table 1: A summary of papers related to affective dialogue systems. TI stands for task-independent; TO stands for tasks-oriented; ST stands for single-turn; MT stands for multi-turn; EA stands for emotion-aware; and EE stands for emotion-expressive.

Authors	Dialogue	Emotion Feat.	Emotion Rep.	Data Source	Keywords
Zhou and Wang [45]	TI,ST	EA+EE	64 Emojis	Tweets	Conditional modeling; Reinforcement Learning
Lubis et al. [46]	TI,MT	EA	Latent Vector	SubTle corpus [47]	Training data processing
Zhou et al. [9]	TI,ST	EA+EE	6-categorical	STC [48]	Modified loss; Auxiliary vocabulary
Peng et al. [49]	TI,ST	EA+EE	VAD	Tweets	Word Embedding;Reinforcement-VAE
Shi and Yu [50]	TO,MT	EA+EE	Multimodal	DSTC-1 [51]	Policy learning
Huang et al. [52]	TI,MT	EA	9-categorical	SubTle [47], CBET [53]	Different fusion stage
Rashkin et al. [54]	TI,MT	EA	32-categorical	ParlAI	Transformer (multi-level attention)
Fung et al. [55]	TO	EA	6-categorical	TED-LIUM [56]	CNN + LSTM
Kong et al. [57]	TI	EA	Binary polarity	Single-turn Tweets	CVAE(g) + CGAN(d)
Niu and Bansal. [58]	TI	EA	Politeness vector	Stanford Politeness Corpus	Fusion; Reinforcement learning
Peng et al. [59]	TI	EA	6-categorical	Chinese dialogue (NLPCC)	Multi-attention; Topic (LDA)
Fung et al. [16]	TO	EA	N.A.	Single-turn Tweets	Memory-Knowledge enhanced

For example, being aware of the emotion in the contexts may implicitly result in generating emotion-expressive responses. The general framework of encoder-decoder can be extended to account for these two sets of features.

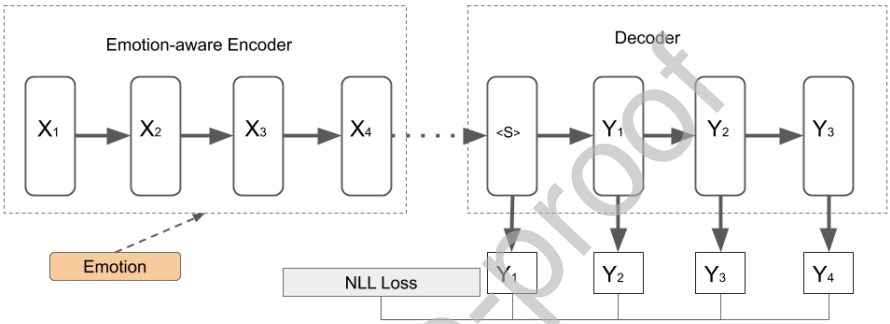
Table 1 summarizes dialogue systems having at least one aforementioned affective features. It shows that most of the work being reviewed are task-independent and have used various representations of emotions. We have also summarized datasets (see Table. 2) used in the references with short descriptions and URL links. These datasets fall into three categories: 1) user-generated conversations (e.g., tweets); 2) crowd-sourcing annotation; 3) simulated data. Moreover, we illustrate typical architectures of vanilla encoder-decoder, emotion-aware encoder, and emotion-expressive decoder in Fig. 9 to facilitate a more straightforward comparison.

3.1. Emotion Analysis

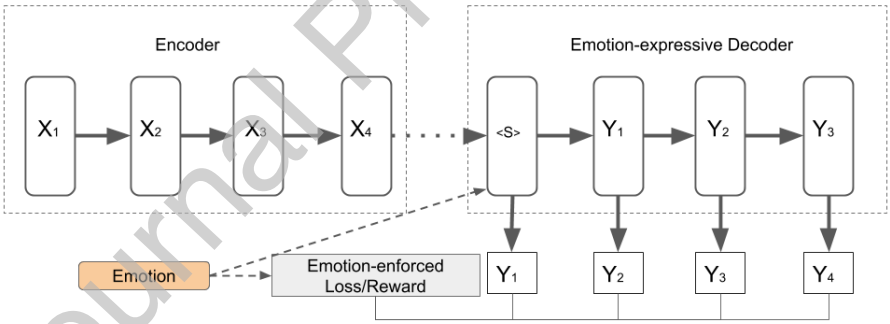
Studies of representing emotion date back to the 19th century, when Charles Darwin proposed his theory of the origin of species. To date, many emotion categorization models have been proposed in the literature [64]. One of the latest is the Hourglass of Emotions [65], a biologically-inspired and psychologically-motivated emotion categorization model for sentiment analysis (Fig. 10). One

Dataset	Description	Download Link	Language
MojiTalk Dataset [45]	Twitter: 596k/32k/32k pairs of original and response	http://drive.google.com/file/d/110fAfxvoNZWv1AWVLeoPZvFZ0Qexr7y0	English
SubTle Corpus [46]	Contact Ameixa for data	http://isemaine-db.eu	English
The STC dataset [48]	Short text conversations in Chinese.	http://github.com/tuxchow/ecm	Chinese
DSTC-1 [51]	Multi-turn dialogues	N/A	English
CBET [53]	labeled 81k short text of 9-category emotion (anger, surprise, joy, love, sadness, fear, disgust, guilt, and thankfulness).	http://github.com/chenyangh/CBET-dataset	English
ParlAI [54]	Multiple, mainly Empathetic Dialog with approx. 24850 dialogues will be on ParlAI	http://parl.ai	English
TED-LIUM [56, 55]	TED-talk monologues	http://openslr.org/7	English
PERSONA-CHAT [60]	Daily dialouge with facts about the speaker.	http://github.com/facebookresearch/ParlAI/tree/master/projects/personachat	English
Stanford Politeness Corpus [61]	User conversation annotated with politeness aspects.	http://cs.cornell.edu/~cristian/Politeness.html	English
ECG NLPCC 2017 Data [62]	Single turn conversation extracted from Chinese Weibo posts.	N/A	Chinese
AIT-2018 [63]	The data comes from SemEval-2018 Task 1: Affect in Tweets	N/A	English

Table 2: List of data-set and resources for Affective Dialogue systems



(a) Emotion-aware encoder.



(b) Emotion-expressive decoder.

Figure 9: Typical architecture of encoder-decoders for affective dialogue generation model.

of the peculiarities of this model is that it describes emotions both in a discrete and in a dimensional form, allowing for the prediction of both categories and intensities of emotions and sentiments [66].

In general, most computational models of emotion only fall into one of these three representation categories: the dimensional approach, the discrete approach, and the appraisal approach [3]. In the dimensional approach, emotions are represented as vectors denoting arousal and valence [67]. The advantage of having a dimensional space is that it allows for measuring the similarity between different emotions [68, 3]. In contrast, the discrete approach classifies emotions into several categories. The number of categories may vary for different settings. For example, Kong et al. [57] modeled the 2-class emotion[1]. Zhou et al. [9] and Peng et al. [59] used 6 classes, while [52] used 9. Rashkin et al. [54] used a more fine-grained 32-class system while Zhou et al. [45] uses 64 emojis. The appraisal approach, finally, studies the links between emotions and elicited cognitive reactions [69].

Another type of emotion representation is distributional, i.e., using embedding to represent an emotion. The advantage of this representation is that emotion types become continuous while interpolation becomes possible. Also, the representation can be directly used as an input into deep learning models. Multimodal emotion information concatenation [50] also falls into this category. The last type of emotion is pragmatic, such as politeness and satisfaction.

With the emotion representation defined, the task of emotion or sentiment analysis is to predict the emotion/sentiment given the sentence or contexts. Namely, given the current utterance x_i, the analyzer predicts the emotion label e_i. The objective can be generally defined to learn the conditional probability distribution $P(e_i|x_i)$. By default, the emotion/sentiment is an attribute attached to the sentence or utterance as a whole which might be seen as an oversimplification. A more fine-grained emotion model assumes one sentence might involve multiple emotion categories regarding different aspects, which is

[1]We consider 2-class sentiment as a type of coarse-grained emotion

Figure 10: The Hourglass of Emotions.

known as aspect-based analysis [70, 71, 72]. For example, "the food is tasty but quite pricey" has expressed a completely opposite sentiment towards #FOOD-QUALITY# and #PRICE#. The objective of an aspect-based emotion analyzer is to learn to predict the emotion labels given both the aspect and sentence using $P(e_i|a_{i,t}, x_i)$, where $a_{i,t}$ denotes the t-th attribute of the current utterance.

In the context of dialogue systems, the task setting may differ from the sentence-alone task. It can be mathematically defined as to predict the emotion label e_i given a sequence of $< x_i, s_i >$ pairs, where $< x_i, s_i >$ represents an utterance and a speaker. Note that, sometimes, the prediction is online so that only dialogue history up to the current time step is visible.

The challenges of analyzing emotions in dialogues extend to several aspects. First of all, the emotion might be expressed in an obscure way, which requires reasoning over contextual information and calls for effective context modeling. Secondly, emotions expressed in dialogues might be highly dependent on the in-

herent and contextual emotional states of both the speaker itself and other parties (inter-personal dependencies). Finally, efficiently blending different modalities, e.g., text, audio, and video, may be required for the continuous interpretation of emotions in a conversation [73] (Fig. 11).

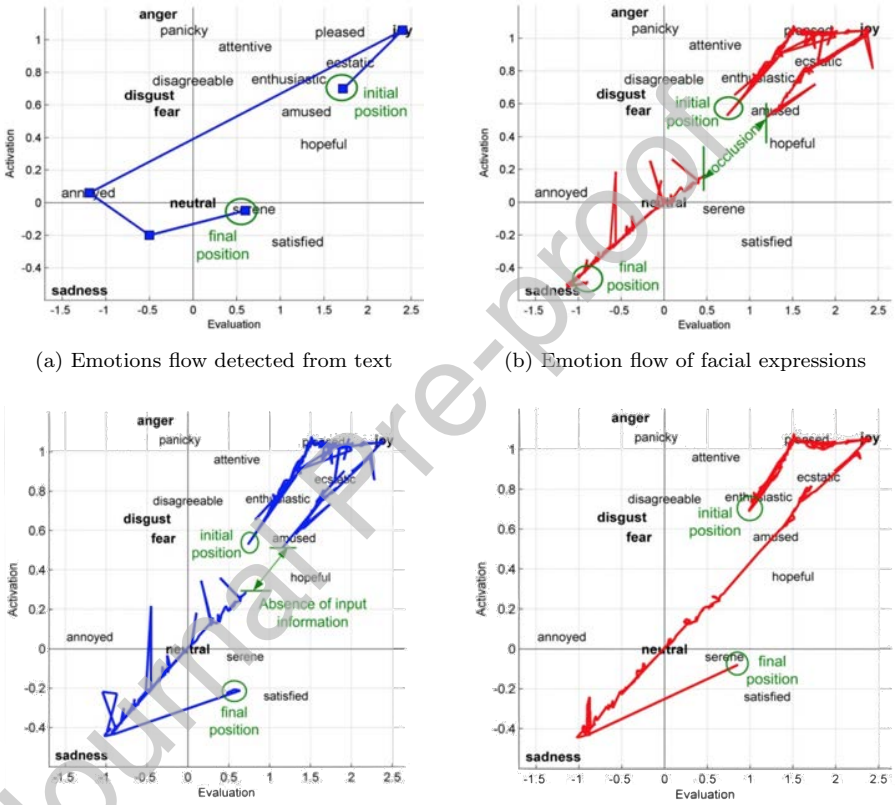
3.2. Emotion-aware Encoders

Emotion-aware encoders differ from general encoders for that the resulting context vector also encodes emotion-related information. As shown in Fig. 9a, there exist three different ways of achieving emotion-awareness depending on whether an emotion label is available or not. If emotion label of context or user input is available, emotion-awareness can be easily achieved by feeding emotion labels as additional features to the encoder [45, 52, 50]. In a more modularized framework, the emotion-aware encoder might also be viewed containing a dialogue management module where the dialogue actions based on emotional states as well as other internal states of a user is modeled as partial observable Markov decision process (POMDP) [28, 29, 74].

On the other hand, it makes sense that emotion labels might be absent from the testing phase. For example, some users do not use explicit emotion markers such as emoji in their inputs. One solution is to have an additional emotion detector that can infer the implicit emotion labels. For example, an additional emotion detector (i.e., emotion classifier) could be employed to recognize the emotion labels [46, 54, 75, 9, 76]. The emotion detector can be trained on either the dialogue training set or supplementary data sets depending on the availability of emotion annotation.

3.3. Emotion-expressive Decoder

Emotion-expressiveness aims at enforcing the generation of emotional responses. As shown in Fig. 9b, it is typically achieved via emotion-specific training of the decoder or directly uses emotion as a controllable variable. We thus refer to such decoders as emotion-expressive decoders. Recent studies have mainly relied on techniques such as CVAE, GAN and reinforcement learning.



(a) Emotions flow detected from text

(b) Emotion flow of facial expressions

(c) Multimodal fusion of Fig.(a) and Fig.(b)

(d) Final (filtered) multimodal fusion

Figure 11: Multimodal emotion flow in an Activation/Evaluation space throughout a sample conversation that went through both positive and negative emotions [73].

The underlying assumption of a controllable emotion variable to the decoder is that there exist a single or multiple latent variables dominating the generation of responses. Some existing work refers to this set of latent variables as latent dialogue states [77]. One natural choice of architecture for modeling the latent dialogue states are CVAEs. Hu et al. [78] proposed that part of the latent dialogue states can be set to controllable variables such as sentiment. In doing so, the encoder is enforced to encode disentangled latent codes and can be seen as a classifier at the same time. On the other hand, the generation of responses is also controlled as these latent codes are feeding to the decoder as inputs.

One problem with CVAE is that it is usually designed to optimize an element-wise loss (i.e., assuming independence of words in the responses). A remedy to such a problem is using GAN framework that converts the learning to a min-max game which additionally optimizes a classification loss based on the overall structure of generated responses. Kong et al. [57] regulated the emotion in generated responses using a discriminator that tells if the response is generated by human beings or the generator providing the emotion label. When maximizing the classification error, the generator is trained to approximate the human-generated responses a simultaneously improves the compatibility of generated response and emotion label.

Although the GAN framework provides an alternative to the element-wise loss, the emotion is used only as an input rather than a learning objective. Yet, a more straightforward way of including emotion is to train the whole framework with emotion-enforced loss or rewards. Namely, design rewards that encourage generating the response with correct emotion, and set penalty for responses with no emotion expressed. However, such intuitive loss may not be differentiable and thus hard to be optimized by the back-propagation. It is then natural to adopt reinforcement learning due to its flexibility in incorporating non-differentiable rewards. More importantly, reinforcement learning allows the framework to learn continuously from user's feedback even after deployment. User's sentiment in such a case might serve as immediate rewards to the system [50, 45, 49].

3.4. Discussion

As mentioned earlier, existing work on affective dialogue systems has mainly focused on either emotion awareness or emotion expressiveness. There are two practical challenges for achieving these two functions:

- Shortage of emotion labels. It arises from the recognition of the emotional states of users. As mentioned in many research work [45, 50, 54], one of the practical challenges in achieving emotion awareness is the lack of human annotation due to the time-consuming annotation process of dialogues. It can be relieved by leveraging weak supervisions instead of ground-truth emotion labels. For example, one could use emotion labels generated by a pre-trained sentiment classifier [9]. Alternatively, multiple data sources could be combined and refined to increase the scale.

- Evaluation of emotion. It is hard to evaluate the user's emotional states as there is sometimes no subtle emotion cue at word level [54] and might exist gaps between the "intrinsic" emotional state of a user, and the one being expressed, and the one being perceived [79]. One solution is to control the process of data collection to guide the user to have desired emotional states and model the "gap" as a noise in data collection.

- Emotion compliance with other goals. The emotion expression should comply with other goals, such as grammatical fluency and naturalness of the generated response. It might be solved by multi-objective optimization and multitask learning [80]. However, it remains an open research problem if the current framework can effectively resole the conflicts between objectives.

- Dependency between the controllable variable and generated words is modeled in the utterance (or turn) level. Not all words should be affected by emotion. For example, function words are mostly only affected by syntax while topical words are dominated by topics. In such a case,

the interaction between emotion variables and the generator should be reformulated to enhance the impacts on the word or phrase level [49].

4. Personalized Dialogue System

The communication between a dialogue system and a human is generally desired to be adaptive to the variance in personal preferences to increase communication effectiveness [97, 98] based on appropriate perception of the speaker's personality of the speaker. On the other hand, personality affects the way of communication in various manners including both linguistic style [99] and acoustic traits [100]. As it feels more natural to interact with a 'thing' that has its own personality, implanting personality into dialogue agents would possibly increase the social attachment [101, 102].

For example, a dialogue system designed to recommend items to a user should have a user preference model so that the recommendation could match the particular user might want to purchase. In this section, we select papers that focus on incorporating personal information into the dialogue system. These papers are sorted out among a wide range of top venues, considering mainly the novelty and impact. We have summarized the list of paper and data set in Table 3 and Table 4.

In short, personalized information of a speaker is the key to precisely perceiving the speaker's intention and inherent states and consequently generate appropriate responses [103]. Efforts have been made towards incorporating personalized features. As shown in Fig. 13, we define the personalized dialogue system (PDS) as a virtual agent having the access to user-specific information without which it might be impossible to generate a proper response adapted to the user's implicit needs. For example, the dialogue agent may respond differently according to the user age group (i.e., old people versus young people) [85], personality (introverts versus extroverts) [104], or gender (woman versus man) [89] that are mostly not part of a visible context.

Authors	Year	Nature of Dialogue	Personalization Features	Personalization Encoder	Personalization Modelling	Backbone Model
Bang et al. [81]	2015	MT;TI	Knowledge-based personalization	RDF triplets	Personality-aware	Traditional NLP methods and matching approach.
Sordoni et al. [82]	2015	ST;TI	Fact-based personalization	BoW	Personality-aware	RNN Language Model + Feed-forward context networks
Li et al. [83]	2016	MT;TI	Identity-based personalization	LSTM	Personality-aware	Seq2Seq with MMI
Al-Rfou et al. [84]	2016	MT;TI	Fact-based personalization	User-embeddings + n-grams embeddings	Personality-aware	Binary logistic regression classifier + Hidden layers with Relu
Joshi et al. [85]	2017	MT;TO	Fact-based + Identity-based personalization	Memory component	Personality-aware	MMN
Zhang et al. [60]	2018	MT;TI	Fact-based personalization	LSTM	Personality-aware	Profile MMN + Attention mechanism
Mo et al. [86]	2018	MT;TO	Fact-based + Identity-based personalization	BoW + belief states	Personality-aware	POMDP-based framework reinforcement learning **POMDP: Partially observable Markov Decision Process*
Qian et al. [87]	2018	MT;TI	Identity-based personalization	GRU	Personality-infused	Binary classifier + MLP
Yang et al. [88]	2018	MT;TI	Fact-based + Identity-based personalization	LSTM	Personality-aware	Twitter LDA + LSTM + Dual Learning with policy gradient
Zheng et al. [89]	2019	MT;TI	Identity-based personalization	2-layer bi-directional RNN with GRU	Personality-infused	RNN + GRU
Zemlyanskiy et al. [90]	2018	MT;TI	Fact-based personalization	LSTM (initialized with GloVe vectors)	Personality-aware	Profile MMN + Attention mechanism
Chu et al. [91]	2018	MT;TI	Fact-based personalization	RNN	Personality-infused	MMN + Skip-thought vectors + Attention mechanism
Olabiyi et al. [92]	2019	MT;TI	Fact-based personalization	RNN	Personality-aware	Hierarchical recurrent encoder-decoder network (HRED) + GANs + Attention mechanism
Luo et al. [93]	2019	MT;TO	Fact-based + Identity-based personalization	Query vector + MMN	Personality-aware	MEMN2N

Table 3: Personalization Dialogue Systems related papers. TI stands for task-independent; TO stands for tasks-oriented; ST stands for single turn; MT stands for multi-turn.

Dataset	Description	Download Link	Language
Twitter FireHose [82]	127M context-message-response triplets from June 2012 to August 2012.	http://microsoft.com/en-us/download/details.aspx?id=52375	English
Twitter Persona Dataset [94]	74,003 users took part in a min of 60 and max 164 conversational turns.	NA	English
Television Series Transcripts [94]	From 2 movies with 13 main characters in a corpus of 69595 turns.	NA	English
Reddit Dataset [84]	2.1 billion messages, 133 million conversations from Reddit web forum from 2007 - 2015	http://reddit.com/r/datasets/comments/3bxlg7	English
Extended bAbI dialog Dataset [85]	Built upon synthetically generated bAbI dialog tasks	http://dropbox.com/s/4i9u4y24pt3paba/personalized-dialog-dataset.tar.gz?dl=1	English
PERSONA-CHAT [60]	Daily dialogue with facts about the speaker.	http://github.com/facebookresearch/ParlAI/tree/master/projects/personachat	English
O2O Coffee d Data [86]	2185 coffee dialogues between 72 consumers and coffee makers from O2O ordering service	N/A	English
Weibo Dataset (WD) [87]	9.6 millions post-response pairs from Weibo.	Upon request	Chinese
Profile Binary Subset (PB) [87]	Extract 76,930 pairs from WD for 6 profile keys and annotated by 13 annotators.	Upon request	Chinese
Profile Related Subset (PR) [87]	Only contains pairs whose posts are positive.	Upon request	Chinese
Weibo Manual Dataset (MD) [87]	600 posts written by 4 human curators	Upon request	Chinese
Twitter Dataset [95]	1.3 million Twitter conversations.	https://homes.cs.washington.edu/$\sim$aritter/twitter_chat	English
Sina Weibo [88]	Chinese Weibo conversations.	N/A	Chinese
PersonalDialog [89]	Multi-turn conversations. Each utterance is associated with a speaker who is marked with traits	N/A	Chinese
Movie Character Trope [91]	CMU Movie Summary dataset provides tropes commonly occurring in stories and media	http://cs.cornell.edu/~cristian/Politeness.html	English
IMDB Dialogue Snippet [91]	Crawled from IMDB Quotes page for each movie.	http://cs.cornell.edu/~cristian/Politeness.html	English
Character Trope Description [91]	The dataset is used for incorporating descriptions of each of the character tropes, which is scraped from TVRopes (http://tvtropes.org)	http://cs.cornell.edu/~cristian/Politeness.html	English
TV Series Transcripts [92]	Extracted from 2 movies with 13 main characters in a corpus of 60595 turns.	N/A	English
Ubuntu Dialog Corpus [96]	Extracted from Ubuntu IRC chat logs.	http://github.com/rkadlec/ubuntu-ranking-dataset-creator	English

Table 4: List of data-set and resources for Personalized Dialogue systems

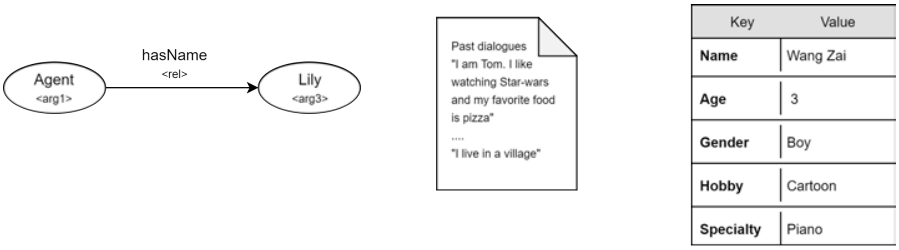


Figure 12: From left to right: Knowledge-based, fact-based and identity-based personalization features.

A typical workflow of PDS reads the current input as well as dialogue history while using an explicit user model to enforce personalized generation of responses. We identify two major components that distinguish a PDS from standard dialogue systems: 1) user modeling and 2) personalized response generation.

4.1. User Modeling

How to represent personality has been the primary concern of many personality theories [104]. Early works have used lexical approaches to describe personality with words or taxonomies [106], which eventually evolved into the well know theory of five-factor model of personality [107]. In addition to the diversity of representation, it is yet challenging to draw a clear picture of the role that personality plays in conversation when considering the gap between "felt" and "perceived" and that between "intentional" and "instinctual" [79]. Existing work have been mostly using oversimplification of personality inspired by different personality theories (as shown in Fig. 12). For example, one may use static features [60] of a speaker, e.g., age group, or narrative facts [83] like "She likes coffee" or triples, $(Lily, speak, French)$ [81]. Based on the classification of personality traits and how it is stored or utilized, we classify the user modeling method into two categories: identity-based and knowledge-based. There are also hybrid systems adopting more than one method of user representation.

4.1.1. Identity-based User Modeling

User modeling is constrained by the form of personal information accessible by the dialogue system. Amongst all types of personal information, the simplest form is the user's identity. Attached to the identity are the static attributes that describe the basic characteristics of the user. The simplicity of these static attributes usually allows low memory consumption so that the indexing and matching process can be very fast. On the other hand, identity-based features are reliable and can be used directly without the need for additional steps of information extraction.

Perhaps, one of the most common sources of identity-based features is the meta-data collected upon registration. Such data collection can not be very comprehensive and, sometimes too coarse-grained which constrains the applicability in various settings. For example, it is fair to state that gender information could help with recommending friends to a user.

As aforementioned, persona facts and identity features may join forces to personalize the response generation [85, 93, 86, 88], thus makes use of both strengths from unstructured (more context-oriented) and structured data (persona-oriented). For example, in restaurant booking, the identity-based features may provide information such as gender, age, or especially favorite food item [85, 93] that is complementary to the description of a user's hobby, which not only helps the system to suggest a suitable restaurant but also facilitates choices of words and speech style.

On the other hand, a range of features have been explored for identity-based user modeling. Notably, to utilize the identity-based features in neural nets, embedding layers are usually employed to map the discrete or continuous features to a dense vector. The vector can simply result from mapping the identities (i.e., user ID) or combined with attribute embeddings [94, 87, 89]. For example, the speaker embedding of [94] has a dialect, register, age, gender which are encoded using LSTM. [87] had agent profile with key-value pairs $< k_{\mathrm{i}}, v_{\mathrm{i}} >$, in which were are attributes such as name, age, gender, hobby, specialty.

4.1.2. Knowledge-based User Modeling

Knowledge-based user modeling uses structured data and predefined rules to match the existing user's information then produce a response. The knowledge-based user modeling is highly correlated with the knowledge-based dialogue system that will be discussed later. In short, a PDS with knowledge-based user modeling can be seen as an instance of a knowledge-based system where the knowledge being used is regarding the user.

As compared with the identity-based user modeling, knowledge-based user modeling is not limited to the meta-data of users. Instead, it could utilize both structured or unstructured information data sources. For example, triples representing the user habits or preferences could be extracted from the user's dialogue history [81]. Sordoni et al. [82] used the bag-of-words (BoW) embeddings of past dialogues of the same user as additional context, which is one of the first attempts to use prior knowledge (dialogue's history from the user) to curate responses. Similarly, Al-Rfou et al. [84] used dialogue histories (context), response, input message, and author (user) as personalization features; then defining user embeddings and the bag of n-gram embeddings to keep track of sentence structures and word order:

$$\psi(M_1, ..., M_n) = \frac{1}{N} \sum_{1 \leq i \leq n} \sum_{w \in ngrams(M_i)} \phi_{ngram}(w)$$

where N is the total number of n-grams from all the messages $\{M_1, ..., M_n\}$. Implementation and data source aside, we think conversation history serves as an anchor to the context which the dialogue system is trying to focus on, and that is the reason why it has been so widely utilized; especially in task-independent dialogue systems where there is no clear goal to achieve but a more specific and meaningful conversation is desired. On the other hand, personality can also be characterized by a set of text description. For example, Mairesse et al. [98] designed a set of questions for rating one's personality.

Personalization might also be a pre-requisite for many tasks. As shown in Fig. 13, the dialogue system requires personal information to make the right recommendation. Personalized reasoning [85] is a task that aims to retrieve

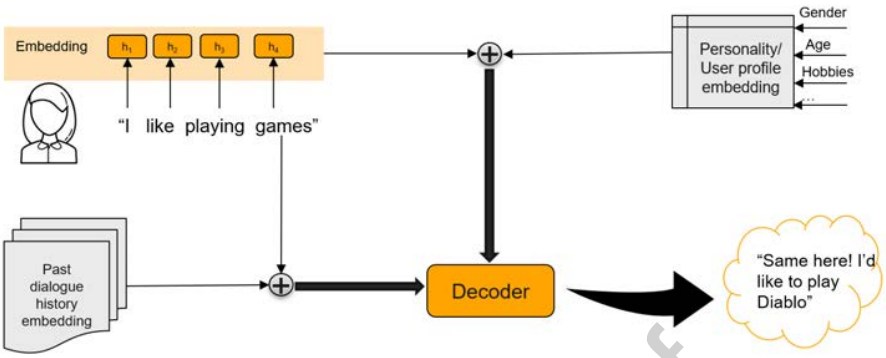


Figure 13: Typical workflow of a personalized dialogue system.

the facts from a knowledge base that is related to a restaurant based on both attributes of the user and the restaurant itself. Conversation history was stored in the memory component [22, 93, 81]. Similarly, Mo et al. [86] also combined both fact-based features (user's utterances and agent's replies) and identity-based features (choices of coffee) for the online coffee shop's dialogue system.

4.2. Personalized Response Generation

With the user modeling representing the personal information of a user, the next key step of a PDS is to generate (or retrieve) the personalized response. Previous works focused on two major approaches: generative methods, which generate appropriate responses during the conversation and retrieval-based methods (or ranking methods), which are capable of selecting suitable responses from a list of candidates or repository. However, the main goal of a PDS is to generate not only suitable but also engaging responses based on prior knowledge of the user. As for affective dialogue systems, we organize our discussion with two sub-topics based on how personalized information is integrated into the decoder (or generator): (1) personality-aware model and (2) personality-infused model to demonstrate the usefulness, capabilities, and limitations of each modeling type.

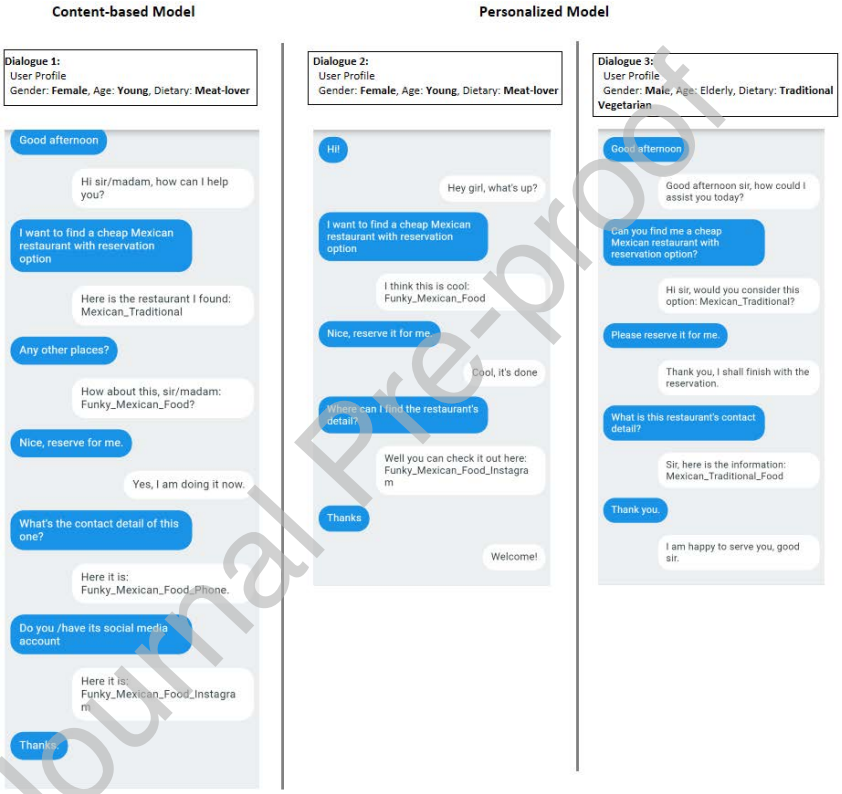


Figure 14: Sample dialogue in a personalized dialogue system.

4.2.1. Personality-aware Model

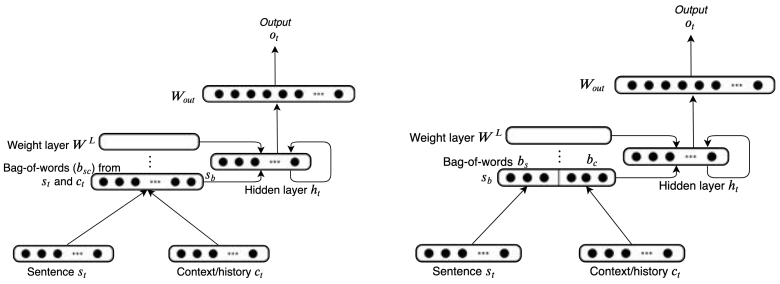
A personality-aware model generates responses adapting to the personality of the user (or other parties of a conversation). In other words, the responses are composed of the awareness of the user's personalized preference. However, what differs significantly from the personality-infused model which will be discussed in the subsequent section is that it does not enforce a personality of the virtual agent itself.

With user modeling, the response can be simply retrieved from a candidate pool [81]. Alternatively, the generator might be taught to generate the response word by word [82] (as illustrated in Fig. 15). For example, it can utilize Seq2Seq model that is both context-sensitive and data-driven. In addition to using input sentence, different context-sensitive information including context c, current message/sentence s_t and response o might be used. Context c represents a sequence of past dialogue exchanges, then the receiver emits message s_t to which the sender reacts by formulating its response, and the estimation of the generated response will be conditioned on past information c and s.

$$p(s_{t+1}|s_1, ..., s_{t1}, c, s_t) = softmax(o_t)$$

These context-sensitive models differ in terms of how they compose the context-message pair (c, s). The context representation does not change through time, hence it forces the context encoder to produce a representation general enough to be useful for generating all words.

Li et al. [94] proposed (1) Speaker model (as illustrated in Fig. 16) and (2) Speaker-Addressee model for personalization task. In (1), they put the Speaker-level vector (which has identity-based features) into the target part of Seq2Seq model. Then they modeled similar words embedding to identify similar speaker embedding. (2) is the Speaker-Addressee model, in which the authors did not only encode the speaker but the addressee as well. For example, if there are two pairs of speaker-addressee $< A, B >$ and $< A', B' >$, with speaker A is similar to A' and B is similar to B'; so A' and B' will have a similar conversation



(a) DCGM I (b) DCGM II

Figure 15: Two different Dynamic-Context Generative models (DCGMs [82]) that make use of past dialogues (context) and current sentence to estimate a personalized response.

and response as A and B, even if they never met before. This can be applied to a knowledge-based dialogue system, as we can now model knowledge like we model the speaker's profile. We can also do reverse mapping for a speaker in the larger set to condition on how it behaves back to the smaller set. One disadvantage is the speaker-addressee model is sensitive to the identity of the addressee, which will often generate sentences that have both the name and the identity of the original addressee.

User profile and conversational history might play a different role in the memory of speakers. Hence, it is reasonable to model them using separate structures. For example, Split Memory [85] extends the original MMN by dividing the memory of the model into two: Profile attributes and Conversation history. The user's attributes are added as separated entries in the profile memory before the dialogue starts, and each dialogue turn is added to the memory. The mechanism of selecting the best responses is the same as the original MMN. The output from both memories are summed element-wise to get the final response. Similarly, Luo et al. [93] gathered the conversations of similar speakers and maintained a global memory in addition to the profile embeddings. On the other hand, this solves the challenge of handling ambiguity among knowledge-based entities, such as the choice between "phone" and "social media", which takes the relation between a speaker profile and knowledge base into account.

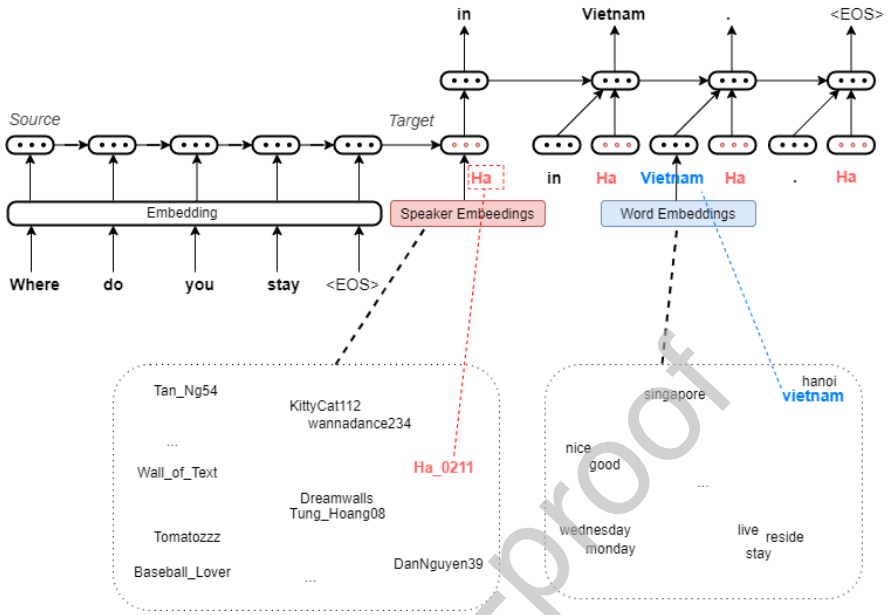


Figure 16: Use of speaker embeddings in encoder-decoder architecture.

Likewise, Zhang et al. [60] handle the history and profile by first performing attention over profiles memory in the first hop, then attention over dialogue histories in the next hops.

Most recently, Olabiyi et al. [92] integrated speaker attributes with a hierarchical recurrent encoder-decoder network (HRED) [108]. The output of the encoder is summarized via attention model and concatenated with the attribute embedding, thus allowing the generator to access the speaker's identity/attributes. The authors also employed the framework of GAN to enforce the awareness of personal information by the generator. Attribute-specific discriminators are used to distinguish responses generated with different user attributes. In doing so, the generator is capable of responding differently to input utterances.

In a large scale setting where numbers of users might present in the system, it can be quite difficult to have enough data for each type of user. It is possible to gather and transfer user knowledge generated in a particular domain or task

setting to a new domain or task. Mo et al. [86] proposed a transfer learning framework for an online coffee-delivery system to model the preferences of different speakers. Its goal is to extend a dialogue system to include a previously unseen concept and then adapt the existing dialogue management system to an extended one [109]. The authors model the personalized Q-function as a general function plus a personal one which has the set of all possible choices of coffees and how it is done (latte, iced, macchiato, etc.), as well as the probability distribution of the speaker's orders. A personal reward point or "punishment" point will be given to the system which depends on the response from the speaker. Yang et al. [88] aim to incorporate personalization criteria. Their framework assumes a source domain D_s and target domain D_t. The model adaptation exploits user-specific information in the personalized dataset (target dataset D_t) to improve the performance of the personalized conversation system by firstly generating an intermediate response and then feed the middle response into the response agent to recover the input.

The main advantages of the transfer learning system include the ability to be trained in both semi-supervised and unsupervised adaptation settings by fine-tuning against different rewards, adequately leverage both paired and unpaired data in the target domain and the capability of exploring user-specific information. However, its disadvantages are large data requirements and need for extensive evaluation. In general, we can use a large collection of general training data as the source domain and the personalized data as a target domain and perform transfer learning from the source domain to the target domain. Reinforcement learning can bypass the exposure bias and non-differentiable issues and maximize future reward in dialogue, thus generates better-personalized responses for different users.

4.2.2. Personality-infused Agent Dialogue Systems

In addition to personality-aware dialogue systems, we have personality-infused agent dialogue systems. Instead of just conditioning on the user's profile or attributes alone, we can assign unique, distinctive personality or profile to an

General Seq2seq model
User: Are you a cat-person or dog-person?
Chatbot: I am a cat-person
User: Are you a dog-person?
Chatbot: Yes, I am a dog-person
Model with personality
User: Are you a cat-person or dog-person?
Chatbot: I am a cat-person, especially the British shorthair
User: Are you a dog-person?
Chatbot: No, I like cats more

Figure 17: Sample responses generated by Seq2Seq and personalized dialogue model.

agent in order to make the conversation smoother, more flexible and natural.

Firstly, as illustrated in Fig. 17, it aims to assign personality and identity to a chat agent for more coherent, natural and realistic responses. The authors use Weibo dataset for the training stage. It has three main components: (1) Profile Detector, (2) Bidirectional Decoder and (3) Position Detector.

The goal of (1) is to select which profile value should be addressed in a generated response (if the user asks about a third-person, not the agent itself i.e.: "How is your sister?", then the agent will go ahead and generate response straight away). This component is a multi-class classifier that uses a multi-layered perceptron (MLP) with the representation of the posts, the weight matrix, and key-values of the profiles. The optimal value is selected with the maximal probability.

Component (2) is a decoder that aims to generate a response in which a profile value will be mentioned. It has a backward decoder and forward decoder but (3) will help to predict the start decoding position. On the other hand, component (3) is designed to provide more supervision to the bidirectional decoder, which is only used during training (because during training the profile values are rarely mentioned in the responses, hence the bidirectional encoder is not aware of which word the decoding should start).

Most importantly, (3) has the ability to alter the training data. This system can provide a model that can generate responses that are coherent to a pre-specified agent profile that does not require learning from dialogue data, and with (3) helps improving performance than a random position picking strategy.

Otherwise, it relies on profiles and requires manual labor works. In conclusion, this paper has the potential to develop more commonsense and better semantic reasoning and can auto-generate many combinations of profiles and improve a lot on a variety of responses.

Secondly, Zhang et al. [89] solved the problem of incorporating specific personality traits into dialogue generation to deliver personalized dialogues. Its Personaldialogue dataset is also from Weibo, which is a multi-turn and each utterance is associated with a speaker who is annotated with traits. In this paper, the problem can be summarized as the following: Given a post $X = x_1, x_2, ..., x_n$ and a set of personality traits $T = t_1, t_2, ..., t_n$, the system should give a response $Y = y_1, y_2, ..., y_n$ that embodies the personality traits in T. As mentioned before, each trait t_i belongs to T is given as a key-value pair $t_i =< k_i, v_i >$.

After having a persona representation, the authors proposed two methods: (1) Persona Aware Attention and (2) Persona Aware Bias. (1) uses persona representation to generate the attention weights at each decoding position (which extends the computation of attention weights used in the decoder) to produce context vector computed at each position that conditioned on the persona representation. In this model, it is more of a personality-aware model. However, in (2), the authors apply directly to estimating the generation distribution by incorporate the representation vector in the output layer of the decoder (which is a 2-layer GRU).

This system can generate responses incorporating certain traits and can choose proper personality traits for the different context, and also tackles the data sparsity issue because the trait representations and dialogue data across speakers are shared. However, it requires a lot of data. The bottom line is this model can integrate richer and subtler traits in the future. It can also learn to choose proper, suitable traits and speech styles or choices of words.

Finally, [91] aimed to predict the character trope based on similar characters across different movies from a database of movie and comic characters' quotes and lines. In the training data, each character will be assigned to different tropes (groups), each group contains several paragraphs describing character-

istics, actions, and personalities, together with character's lines in a dialogue (prior knowledge).

The authors proposed Attentive Memory Network (AMN), which consists of Attentive Encoders (1) and Knowledge-Store Memory Module (2). (1) has Attentive Snippet Encoder (individual level) to capture the features and the relevance of each word in the given text, and Attentive Inter-Snippet Encoder (across multiple snippets) captures the inter-snippet relationship. The attention mechanism is used to assign low attention scores to irrelevant snippets. Then (1) will result in the summary vector, which is also the persona representation where it captures both internal features and external features (contextual information, other characters' lines). Meanwhile, (2) incorporates description with skip-thought vectors [110] to initialize memory keys, and the values are learnable embeddings of trope categories. The final step is to calculate similarities between persona representation and keys.

In conclusion, this paper demonstrates that the use of a multilevel attention mechanism greatly outperforms a baseline GRU model, about the character trope classification task that serves as a testbed for learning and assigning personas from dialogues.

5. Knowledge-based Dialogue System

Generating a conversation is a process of searching and communicating with the knowledge that might come from multiple sources including the current dialogue, personal background, or even external knowledge sources such as a knowledge graph [111]. The comprehension of dialogue thus requires access to the background knowledge which has created a gap between responses generated by human beings and those by data-driven dialogue agents [13, 48, 18]. Fig. 18 shows an example in which the external knowledge plays an important role in inferring the appropriate word in response. It can be seen that without knowing "Universal studios" is at Sentosa and can be shortened as "USS", the generator is not able to infer such information from the given input. Therefore, given the

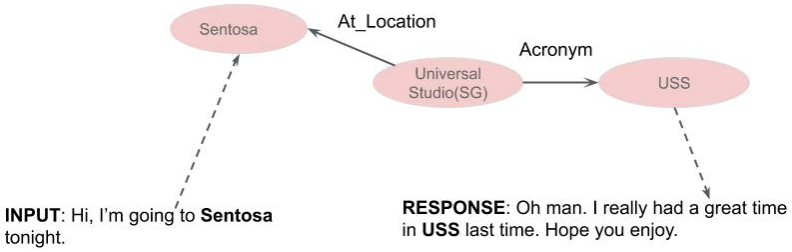


Figure 18: Example of external knowledge aiding dialogue generation.

need for modeling knowledge in dialogues, we refer to those dialogue systems with explicit access and modeling of external knowledge that are not visible in the current dialogue as a knowledge-based dialogue system.

Extending the basic architecture of encoder-decoder, a knowledge-based dialogue system typically has two additional components: 1) a knowledge encoder (see Section 5.1) encoding the knowledge into some sorts of representation and 2) a knowledge-aware decoder (see Section 5.2) generating responses conditioned on both the context and external knowledge. Based on a set of critical dimensions, we summarize all papers and datasets having been discussed in this section and related to incorporating into Table 5 and Table 6.

It is hard to come up with a universal encoder because of the heterogeneous nature of knowledge. Instead, we classify knowledge encoders into two main categories based on if the knowledge to be encoded is structural or not.

5.1. Knowledge Encoding

5.1.1. Structured Knowledge

Structured knowledge plays an essential role in language understanding for their well-defined forms and simplicity in use. One of the criteria of a comprehensive and structured knowledge base is that it should be sufficiently general to account for a large number of concepts and their relations, and workable with real-world settings [128]. Efforts on building a structured knowledge base can be dated back to the 1950s [129] where the knowledge is represented in a set of rules expressing permissible transformation operations. Later, the structured

Authors	Year	Forms of knowledge	Knowledge encoder	Backbone model	Nature of dialogue	Domain	Main Challenge
Lowe et al. [96]	2015	Unstructured knowledge	LSTM	LSTM + Encoder-Decoder	Multi turns; Task-independent	Operation System	Incorporate unstructured knowledge into dialogue generation.
He et al. [112]	2017	Structured knowledge	Recursive graph embedding; Copy + attention	LSTM + Encoder-Decoder	Multi-turn; Task-dependent; Collaborative	Mutual Friends	Represent structured knowledge. Condition the generation of response on both knowledge and dialogue history
Madotto et al. [113]	2018	Structured knowledge	BoW embedding; Copy + attention + sentinel	LSTM + Encoder-Decoder	Multi-turn; Task-dependent	Multi-domain	Represent structured knowledge. Condition the generation of response on both knowledge and dialogue history.
Guo et al. [114]	2018	Structured knowledge	Sequential action encoder	GRU + Encoder + Grammar-guided Decoder	Multi-turn; Task-independent; Q&A	Open-domain	Jointly model searching and inference on knowledge. dialogue memory management by inferring a set of logic forms.
Wu et al. [115]	2019	Structured knowledge	BoW embedding + MMN; Multi-hop read and write.	GRU + Global Mem Encoder + Local Mem Decoder	Multi-turn; Task-dependent	Multi-domain	Knowledge base used to be very large and dynamic and thus difficult to be incorporated into a learning framework.
Xu et al. [116]	2019	Structured knowledge	Directed probabilistic graph	DQN +Relation matrix	Multi-turn; Task-dependent	Medical	Modelling dialogue rationality in the context of medical knowledge and symptom-disease relationsMismatch between template-based dialogue and real-world scenarios.
Ghazvininejad et al. [117]	2018	Unstructured knowledge	LSTM encoder	GRU + Seq2Seq	Multi-turn, task-independent	Open domain	Incorporate unstructured knowledge
Liu et al. [118]	2018	Structured	Summary encoding (count of entities)	LSTM + Policy network	Multi-turn; task-dependent	Restaurant search;Movie booking	Tune the pre-trained model using online user feedback; Resolve mismatch of dialogue state distribution between offline and online modes
Reddy et al. [119]	2019	Structured	BoW embedding + multi-level attention + Copy	GRU + hierarchical Encoder + Decoder	Multi-turn;Task-dependent	Restaurant search;In-car;Movie booking	Knowledge tuples are large in amount which might cause memory explosion. Knowledge triples enforces inference of relation between attributes (considering indirect links). Enforces a common reader for two different types of data (dialogue history and knowledge)
Zhou et al. [120]	2018	Structured	Triplet Embedding + Hierachical Graph Attention	GRU + hierarchical Encoder + Decoder	Single turn; Chit-chat	Open domain	Enhance language understanding with commonsense knowledge.

Table 5: Summary of work involving an external knowledge component

Dataset	Description	Download Link	Language
Foursquare + Twitter [117]	Comments left by customers about restaurants. 23M general dataset of 3-turn conversations grounded with Foursquare tips. Tweets with handles found in Foursquare	N/A	English
CSQA [114]	Created from Wikipedia. Questions are classified as kinds of types (e.g., simple questions or logical reasoning).	http://amritasaha1812.github.io/CSQA	English
Google Simulated Dataset [118]	GST: Dataset of conversations between an agent and a simulated user.	http://github.com/google-research-datasets/simulated-dialogue	English
MZ Medical Dialogue [121]	Dialogues collected from Baidu Muzhi Doctor website.	http://sdspeople.fudan.edu.cn/zywei/data/acl2018-mds.zip	Chinese
DX Medical Dialogue [116]	Dialogues collected from another Chinese medical consulting web (dxy.com).	N/A	Chinese
Stanford In-car Assistant Dataset [122]	Multi-domain dialog dataset collected using Crowd Sourcing.	http://nlp.stanford.edu/blog/a-new-multi-turn-multi-domain-task-oriented-dialogue-dataset	English
bAbI Dialog Dataset [123]	Task-specific dialogues in the restaurant domain	http://github.com/MILVLG/bottom-up-attention.pytorch	English
ARC [124]	This consists of a collection of scientific questions and a large scientific text corpus containing a large number of scientific facts.	http://data.allenai.org/arc/arc-corpus	English
SEMEVAL 2018 - task 11 [125]	Questions describing events about daily activities.	http://competitions.codalab.org/competitions/17184	English
CamRes [126]	Data on restaurant searching dialog for restaurants in the Cambridge area.	http://github.com/shawnmun/NNDIAL/blob/master/data/CamRest676.json	English
Ubuntu Dialog Corpus [96]	Extracted from Ubuntu IRC chat logs.	http://github.com/rkadlec/ubuntu-ranking-dataset-creator	English
CoCoA Dataset [112]	This dataset was created by crowdsourcing.	http://stanfordnlp.github.io/cocoa/	English
Maluuba Frames [127]	Task oriented dialogues for hotel and flight booking.	http://datasets.maluuba.com/Frames	English

Table 6: List of data-set and resources for knowledge-based dialogue systems

knowledge became the backbone of expert systems [130] which is usually specific to a particular domain or task. However, the development and maintenance of these knowledge base systems are generally relying on human efforts and thus not scalable and hard to be generalized to other domains. Additionally, commonsense computing [131] aims to develop large-scale commonsense knowledge bases, e.g., ConceptNet [132] and SenticNet [133], that can be automatically built and updated over time.

Thanks to the rapid growth of large scale knowledge bases [134], structured knowledge is deemed as a main source for the dialogue agent to obtain relational information of different kinds of entities, e.g., named entities but also time expressions [135], which assists in understanding the semantics of language. In the light of recent progress in deep neural network, embedding-based knowledge encoder has been adopted by existing frameworks. It can be represented as a set of relation triplets, i.e., $\{(e_1, r, e_2)\}$, where e_1 and e_2 are the two entities and r being the relation type. For example, ("Jimmy's House", Is_A, "Friend's Home") simply means that "Jimmy's house is my friend's home". Since entities and relations are all consisting of words, one of the most straightforward choices of an encoder is the BoW encoder [113, 115]. At first, each word of entities and relations is mapped to a dense vector via looking up in an embedding table. All the word vectors are then summed (or averaged) over to get a single vector for the triplet.

One problem with the BoW representation is that the structural information is lost to some extent. That said, it makes no difference whether a word appears in e_1, r, or e_2. A more accurate encoder takes into consideration the word order by using a sequential model (e.g., an RNN) [136] or separated parameters for the head, relation, and tail [120]. Moreover, the triplet embedding considers only one triplet at a time and fails to account for the global structure of knowledge. Reddy et al. [119] directly represent the data cell (containing multiple attributes and values) by a set of key-value pairs, while He et al. [112] proposed a recursive graph embedding where information can propagate from neighbors to the entities of interest.

5.1.2. Unstructured Knowledge

Perhaps, the biggest drawback of using structured knowledge in building dialogue systems is that the information required by the dialogue system may sometimes not be aligned with the structure of knowledge. Moreover, the process of structuring knowledge involves either human efforts or rule-based filters that are performed as a separate step which is typically not optimized together with the dialogue model. In comparison to structured knowledge, unstructured knowledge might be less constrained, available on a much larger scale, and contain richer information yet to be filtered. Since unstructured knowledge is usually in the form of plain texts, the suitable knowledge encoders for unstructured knowledge are mostly sequence encoders [96, 117] that were used to convert a sequence of words into a dense vector. Moreover, unstructured knowledge encoders can be trained end-to-end with the rest of the dialogue system to achieve optimal performance.

5.2. Knowledge-aware Decoding

Although some particular types of encoded knowledge (e.g., a count vector) might be simply passed to the decoder as additional input, it is still the biggest challenge of modeling in dialogue. This section will discuss how knowledge could be utilized to facilitate the generation of more informative and coherent responses. Generating responses typically involves two types of knowledge sources. The first one is from the history of dialogue, including utterances of previous turns as well as the user's input at the current dialogue turn. We refer to this knowledge source as the historical knowledge, while there is another source of knowledge to be embedded into the response, referred to as the predictive knowledge.

5.2.1. Knowledge Attention

The knowledge encoder converts the knowledge base into some representation. As seen from the previous section, the most popular representation is a form of dense vectors called knowledge embedding. Given the context and the

set of knowledge embedding, it needs to read or retrieve the relevant knowledge that will be then used for conditioning the generation of a response.

The retrieval and searching process of knowledge can be done in a heuristic fashion, especially when the knowledge base is on a large scale. Lowe et al. [96] retrieved unstructured texts using a combination of TD-IDF and hashing. One popular and probably more effective alternative dealing with a smaller (or refined) set of knowledge is to compute a weighted sum over the knowledge embedding. As we have discussed in the previous section on the attention model, the weights of subsets of knowledge bases (e.g., triples) can be obtained by computing an attention vector. As for the attention over a sequence of words, each encoded knowledge vector is passed to a MLP together with a query vector representing to current dialogue states. Zhou et al. [120] proposed to learn hierarchical attention where the first layer is over the knowledge graphs while it has second layer attention over the triples.

5.2.2. Copy

Two extensions of attention mechanism playing an essential role in end-to-end dialogue generation are pointer network [137] and CopyNet [138]. The main idea is to use the attention as pointers to select and copy words from the input. It was first proven by Eric et al. [139] that augmenting the standard Seq2Seq framework with the copy mechanism could outperform retrieve-based methods. Later, Eric et al. [122] propose adding to the vocabulary distribution a copy distribution over knowledge base entries. In other words, a word can be generated by either emitting from the distribution over a fixed vocabulary or by copying a word from the knowledge base. Madotto et al. [113] introduced a memory augmented neural architecture with a differentiable external memory component storing the dialogue history as well as triples from a knowledge base. It differs from standard CopyNet in the sense that copy distribution is trained separately to minimize the loss on the heuristically generated ground truth pointers. Instead of using only one pointer to filter the knowledge and dialogue history, Wu et al. [115] apply a global-to-memory pointer with a global

memory to filter the knowledge and context and a local memory at the decoder side to copy words dynamically.

In addition to architectures working with triple embeddings, there also exist approaches to tackle with more structured knowledge representation. Reddy et al. [119] apply a multi-level memory architecture to learn pointers to values hierarchically grouped by query, result, and attributes. Guo et al. [114] model the inference of logical forms based on the current dialogue context as a sequence of actions. Copying a sub-sequence of actions is then considered as a particular type of action in decoding.

6. Future Directions

Many research challenges remain in the context of empathetic dialogue systems. For example, little effort has been devoted to combine the three key components (i.e., personalization, knowledge, and emotion) to build a more comprehensive empathetic system. With advances in each subtopic, it becomes possible to further extend this research area on different fronts:

1. **Multi-goal Management** As pointed out by Pollack et al. [140], communication might be overloaded with multiple objectives. This becomes even more true when emotion, personality and knowledge are fused into the system. The dialogue agent should take into account all different aspects, exhibiting perception of the user's inherent states, communicating information, and minimizing the communicative efforts. The problem then becomes how to effectively search for an optimal solution to incorporate and optimize these objectives simultaneously.
2. **Explicit Affective Policy** Existing literature have used emotion to affect the choice of action (or implicit action in an end-to-end framework). However, emotion can be considered as explicit actions in the action space to display the affective behavior more straightforwardly. For example, a virtual agent could take different strategies for parallel empathy (mirroring the other one's emotion) and reactive empathy (providing insight to

recover from other one's emotional states) [39].

3. **Long-term Empathy Modeling** The display of empathy in dialogues is usually engaged in long-term activities. A reliable model that can claim to have learned to be empathy must be based on analysis of long-term data. Also, the three main components – emotion, personality and knowledge are of both static and dynamic features. In other words, they have stable bases while also being subject to changes. It remains an open research and engineering challenge to build a framework that is capable of engaging users in long-term dialogue data collection and continuously develop the conversational model to adapt to changes.
4. **Dialogue Generation with Target-dependent Emotion** Although emotion has been taken into consideration by the dialogue generation model, existing work has omitted the dependency of emotion and target. In other words, emotion has been assumed to be a uni-dimensional variable without considering it may be specified towards different targets. A similar problem has been raised for emotion or sentiment classification [141, 142, 143], but has been missing in the context of dialogue systems. A further study in this direction would be to combine target-dependent emotion with user modeling, as emotion is a particular dimension attached to the speaker and other participants of the conversation. The emotion and personality should be two correlated dimensions of the speaker, and thus should be jointly modeled.
5. **Dialogue Generation with Emotion Knowledge**. An existing knowledge base might contain sentimental or emotional knowledge, e.g., SenticNet, that can help to recognize the emotional states of the speaker and understand background information beyond the context. On the other hand, such knowledge could also help generate emotion-coherent responses.
6. **Incorporate Cues from Multimodal Input** The goal of modeling empathy in conversation is to personify the dialogue system. However, inspired by the fact that communication between humans could be multimodal, the output of a dialogue system could be extended to multiple

modalities to make it more empathetic. Existing work has shown that multimodality helps to improve the accuracy of emotion detection from dialogue [144, 145, 146, 147, 148]. In the future, it would be worth investigating whether empathy in dialogues could be enhanced by considering more input channels such as audio signals and body gestures.

7. **Personalized Diversifying Dialogue Generation** One of the biggest advantages brought by personalization is the diversity of responses. Namely, the responses generated or retrieved can be customized for a given user profile. Indirectly, this diversifies the responses generated. However, existing work ignores the fact that the model might always generate similar responses to the same group of users. One research problem is how to encourage the intra-group diversity of personalized dialogues.
8. **Deeper Conversation and User modeling** Most dialogue systems available on the market today approach conversations between user and chatbot merely as an information retrieval problem. Given a query from the user, their main goal is simply to retrieve the answer that is statistically more plausible. In the future, dialogue systems will have to do much more than that, e.g., create a model of each conversation, understand how user's emotions change throughout the conversation, remember prior conversations and preferences of the users, understand user's needs and intentions [103]. This can be made possible by both applying more 'semantics-aware' deep learning techniques, e.g., capsule networks [149], but also by deconstructing the problem into all the relevant subtasks involved in conversation understanding, e.g., sarcasm detection [80], time expression and named entity recognition [135], anaphora resolution [150], microtext normalization [151], and more.

7. Conclusion

Although emotion, personality and knowledge have been considered key components by existing research on dialogue systems, little work has been done to-

wards investigating the correlation between them in a broader context in order to enhance human-computer interaction. In this survey, we provided a unified view of these different research efforts under the topic of empathetic dialogue systems and discussed recent advancements and trends in this context. As one of the key features in next-generation dialogue systems, empathetic features are imposing more challenges to this research domain. Researchers have explored a variety of settings and problems related to empathy and have attained successful results. The road to emulating human-to-human conversations, however, is still long and bumpy. For each of the three sub-topics, we surveyed the most recent and representative work and outlined a logical storyline for ease of comprehension. Finally, we identified a few promising future directions for this exciting research area that could one day become the killer application of conversational artificial intelligence.

8. Acknowledgements

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

[1] R. S. Nickerson, S. F. Butler, M. Carlin, Empathy and knowledge projection, The social neuroscience of empathy (2009) 43–56.

[2] K. Liu, R. W. Picard, Embedded empathy in continuous, interactive health assessment, in: CHI Workshop on HCI Challenges in Health Assessment, Vol. 1, Citeseer, 2005, p. 3.

[3] M. F. McTear, Z. Callejas, D. Griol, The conversational interface, Vol. 6, Springer, 2016.

[4] R. Looije, M. A. Neerincx, F. Cnossen, Persuasive robotic assistant for health self-management of older adults: Design and evaluation of social be-

haviors, International Journal of Human-Computer Studies 68 (6) (2010) 386–397.

[5] J. Weizenbaum, Computer Power and Human Reason: From Judgment to Calculation, W H Freeman & Co, 1976.

[6] J.-Y. Magadur, F. Gavignet, F. Andry, F. Charpentier, A french oral dialogue system for flight reservations over the telephone, in: Third European Conference on Speech Communication and Technology (EUROSPEECH), 1993, pp. 1789–1792.

[7] F. Morbini, E. Forbell, D. DeVault, K. Sagae, D. R. Traum, A. A. Rizzo, A mixed-initiative conversational dialogue system for healthcare, in: SIGDIAL Conference, Association for Computational Linguistics, 2012, pp. 137–139.

[8] A. Khatua, E. Cambria, A. Khatua, I. Chaturvedi, Let's chat about brexit! a politically-sensitive dialog system based on twitter data, in: ICDM Workshops, 2017, pp. 393–398.

[9] H. Zhou, M. Huang, T. Zhang, X. Zhu, B. Liu, Emotional chatting machine: Emotional conversation generation with internal and external memory, in: AAAI Conference on Artificial Intelligence, 2018, pp. 730–739.

[10] G. G. Lee, H. K. Kim, M. Jeong, J.-H. Kim, Natural language dialog systems and intelligent assistants, Springer, 2015.

[11] H. Xu, H. Peng, H. Xie, E. Cambria, L. Zhou, W. Zheng, End-to-end latent-variable task-oriented dialogue system with exact log-likelihood optimization, World Wide Web (2020) .

[12] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao, Deep learning based text classification: A comprehensive review, arXiv preprint arXiv:2004.03705.

[13] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.

[14] H. Chen, X. Liu, D. Yin, J. Tang, A survey on dialogue systems: Recent advances and new frontiers, ACM SIGKDD Explorations Newsletter 19 (2) (2017) 25–35.

[15] J. Gao, M. Galley, L. Li, et al., Neural approaches to conversational AI, Foundations and Trends in Information Retrieval 13 (2-3) (2019) 127–298.

[16] P. Fung, D. Bertero, P. Xu, J. H. Park, C.-s. Wu, A. Madotto, Empathetic dialog systems, in: Language Resources and Evaluation Conference (LREC), 2018.

[17] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, S. Khudanpur, Recurrent neural network based language model., in: INTERSPEECH, ISCA, 2010, pp. 1045–1048.

[18] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.

[19] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[20] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473.

[21] W. Zaremba, I. Sutskever, Learning to execute, arXiv preprint arXiv:1410.4615.

[22] J. Weston, S. Chopra, A. Bordes, Memory networks, CoRR abs/1410.3916.

[23] D. P. Kingma, M. Welling, Auto-encoding variational bayes, in: International Conference on Learning Representation, 2013.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved training of wasserstein gans, in: Advances in neural information processing systems, 2017, pp. 5767–5777.

[25] B. Zhang, D. Xiong, H. Duan, M. Zhang, et al., Variational neural machine translation, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 521–530.

[26] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, Cambridge, MA, USA, 2014, pp. 2672–2680.

[27] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, J. Gao, Deep reinforcement learning for dialogue generation, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 1192–1202.

[28] T. H. Bui, M. Poel, A. Nijholt, J. Zwiers, A tractable hybrid ddn–pomdp approach to affective dialogue modeling for probabilistic frame-based dialogue systems, Natural Language Engineering 15 (2) (2009) 273–307.

[29] T. H. Bui, J. Zwiers, M. Poel, A. Nijholt, Affective dialogue management using factored pomdps, in: Interactive Collaborative Information Systems, Springer, 2010, pp. 207–236.

[30] V. Rieser, O. Lemon, Reinforcement learning for adaptive dialogue systems: a data-driven methodology for dialogue management and natural language generation, Springer Science & Business Media, 2011.

[31] S. Marsella, J. Gratch, Computationally modeling human emotion, Communications of the ACM 57 (12) (2014) 56–67.

[32] C. Marinetti, P. Moore, P. Lucas, B. Parkinson, Emotions in Social Interactions: Unfolding Emotional Experience, 2011, pp. 31–46. doi:10.1007/978-3-642-15184-2_3.

[33] K. R. Scherer, What are emotions? and how can they be measured?, Social Science Information 44 (4) (2005) 695–729. arXiv:https://doi.org/10.1177/0539018405058216, doi:10.1177/0539018405058216.
URL https://doi.org/10.1177/0539018405058216

[34] K. Boehner, R. DePaula, P. Dourish, P. Sengers, How emotion is made and measured, Int. J. Hum.-Comput. Stud. 65 (4) (2007) 275–291. doi:10.1016/j.ijhcs.2006.11.016.
URL https://doi.org/10.1016/j.ijhcs.2006.11.016

[35] N. H. Frijda, Emotion, cognitive structure, and action tendency, Cognition and emotion 1 (2) (1987) 115–143.

[36] J. R. Busemeyer, E. Dimperio, R. K. Jessup, Integrating emotional processes into decision-making models, Integrated models of cognitive systems 1 (2007) 213.

[37] R. Beale, C. Creed, Affective interaction: How emotional agents affect users, International Journal of Human-Computer Studies 67 (9) (2009) 755 – 776. doi:https://doi.org/10.1016/j.ijhcs.2009.05.001.
URL http://www.sciencedirect.com/science/article/pii/S1071581909000573

[38] P. Dybala, M. Ptaszynski, R. Rzepka, K. Araki, Activating humans with humor–a dialogue system that users want to interact with, IEICE TRANSACTIONS on Information and Systems 92 (12) (2009) 2394–2401.

[39] C. N. Moridis, A. A. Economides, Affective learning: Empathetic agents with emotional facial and tone of voice expressions, IEEE Transactions on Affective Computing 3 (3) (2012) 260–272.

[40] R. W. Picard, Affective computing, MIT press, 2000.

[41] R. W. Picard, J. Klein, Computers that recognise and respond to user emotion: theoretical and practical implications, Interacting with Computers 14 (2) (2002) 141–169.

[42] I. A. Iurgel, A. F. Marcos, Employing personality-rich virtual persons—new tools required, Computers & Graphics 31 (6) (2007) 827 – 836.

[43] J. Pittermann, A. Pittermann, W. Minker, Handling Emotions in Human-Computer Dialogues, 1st Edition, Springer Publishing Company, Incorporated, 2009.

[44] S. Pauletto, B. Balentine, C. Pidcock, K. Jones, L. Bottaci, M. Aretoulaki, J. Wells, D. P. Mundy, J. Balentine, Exploring expressivity and emotion with artificial voice and speech technologies, Logopedics Phoniatrics Vocology 38 (3) (2013) 115–125.

[45] X. Zhou, W. Y. Wang, Mojitalk: Generating emotional responses at scale, in: Annual Meeting of the Association for Computational Linguistics (ACL), 2018, pp. 1128–1137.

[46] N. Lubis, S. Sakti, K. Yoshino, S. Nakamura, Eliciting positive emotion through affect-sensitive dialogue response generation: A neural network approach, in: AAAI Conference on Artificial Intelligence, 2018, pp. 5293–5300.

[47] D. Ameixa, L. Coheur, P. Fialho, P. Quaresma, Luke, i am your father: dealing with out-of-domain requests by using movies subtitles, in: International Conference on Intelligent Virtual Agents, Springer, 2014, pp. 13–21.

[48] L. Shang, Z. Lu, H. Li, Neural responding machine for short-text conversation, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Vol. 1, 2015, pp. 1577–1586.

[49] D. Peng, M. Zhou, C. Liu, J. Ai, Human–machine dialogue modelling with the fusion of word-and sentence-level emotions, Knowledge-Based Systems 192 (2020) 105319.

[50] W. Shi, Z. Yu, Sentiment adaptive end-to-end dialog systems, in: Annual Meeting of the Association for Computational Linguistics (ACL), 2018, pp. 1509–1519.

[51] A. Raux, B. Langner, D. Bohus, A. W. Black, M. Eskenazi, Let's go public! taking a spoken dialog system to the real world, in: Ninth European conference on speech communication and technology, 2005.

[52] C. Huang, O. R. Zaïane, A. Trabelsi, N. Dziri, Automatic dialogue generation with expressed emotions, in: NAACL-HLT, 2018, pp. 49–54.

[53] A. Yadollahi, A. G. Shahraki, O. R. Zaiane, Current state of text sentiment analysis from opinion to emotion mining, ACM Computing Surveys (CSUR) 50 (2) (2017) 25.

[54] H. Rashkin, E. M. Smith, M. Li, Y.-L. Boureau, I know the feeling: Learning to converse with empathy, in: ArXiv preprint, 2018, p. CoRR abs/1811.00207.

[55] P. Fung, A. Dey, F. B. Siddique, R. Lin, Y. Yang, D. Bertero, Y. Wan, R. H. Y. Chan, C.-S. Wu, Zara: A virtual interactive dialogue system incorporating emotion, sentiment and personality recognition, in: International Conference on Computational Linguistics (COLING), 2016, pp. 278–281.

[56] A. Rousseau, P. Deléglise, Y. Esteve, Enhancing the ted-lium corpus with selected data for language modeling and more ted talks., in: LREC, 2014, pp. 3935–3939.

[57] X. Kong, B. Li, G. Neubig, E. H. Hovy, Y. Yang, An adversarial approach to high-quality, sentiment-controlled neural dialogue generation, in: ArXiv preprint, 2019, p. CoRR abs/1901.07129.

[58] T. Niu, M. Bansal, Polite dialogue generation without parallel data, Transactions of the Association for Computational Linguistics 6 (2018) 373–389.

[59] Y. Peng, Y. Fang, Z. Xie, G. Zhou, Topic-enhanced emotional conversation generation with attention mechanism, Knowledge Based Systems 163 (2019) 429–437.

[60] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, J. Weston, Personalizing dialogue agents: I have a dog, do you have pets too?, arXiv preprint arXiv:1801.07243.

[61] C. Danescu-Niculescu-Mizil, M. Sudhof, D. Jurafsky, J. Leskovec, C. Potts, A computational approach to politeness with application to social factors, in: Proceedings of ACL, 2013.

[62] X. Huang, J. Jiang, D. Zhao, Y. Feng, Y. Hong, Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings, Vol. 10619, Springer, 2018.

[63] S. Mohammad, F. Bravo-Marquez, M. Salameh, S. Kiritchenko, SemEval-2018 task 1: Affect in tweets, in: Proceedings of The 12th International Workshop on Semantic Evaluation, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 1–17.

[64] Z. Wang, S. Ho, E. Cambria, A review of emotion sensing: Categorization models and algorithms, Multimedia Tools and Applications (2020) .

[65] Y. Susanto, A. Livingstone, B. C. Ng, E. Cambria, The hourglass model revisited, IEEE Intelligent Systems 35 (5) (2020) .

[66] M. S. Akhtar, A. Ekbal, E. Cambria, How intense are you? predicting intensities of emotions and sentiments using stacked ensemble, IEEE Computational Intelligence Magazine 15 (1) (2020) 64–75.

[67] M. Soleymani, J. Lichtenauer, T. Pun, M. Pantic, A multimodal database for affect recognition and implicit tagging, IEEE Transactions on Affective Computing 3 (1) (2012) 42–55.

[68] E. Cambria, J. Fu, F. Bisio, S. Poria, AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis, in: AAAI, 2015, pp. 508–514.

[69] S. Marsella, J. Gratch, P. Petta, et al., Computational models of emotion, A Blueprint for Affective Computing-A sourcebook and manual 11 (1) (2010) 21–46.

[70] S. Poria, I. Chaturvedi, E. Cambria, F. Bisio, Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis, in: IJCNN, 2016, pp. 4465–4473.

[71] Y. Ma, H. Peng, T. Khan, E. Cambria, A. Hussain, Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis, Cognitive Computation 10 (4) (2018) 639–650.

[72] H. Peng, Y. Ma, Y. Li, E. Cambria, Learning multi-grained aspect target sequence for chinese sentiment analysis, Knowledge-Based Systems 148 (2018) 167–176.

[73] E. Cambria, N. Howard, J. Hsu, A. Hussain, Sentic blending: Scalable multimodal fusion for continuous interpretation of semantics and sentics, in: IEEE SSCI, Singapore, 2013, pp. 108–117.

[74] D. Li, Y. Li, S. Wang, Interactive double states emotion cell model for textual dialogue emotion prediction, Knowledge-Based Systems 189 (2020) 105084.

[75] D. Griol, J. M. Molina, Z. Callejas, Modeling the user state for context-aware spoken interaction in ambient assisted living, Applied intelligence 40 (4) (2014) 749–771.

[76] S. W. McQuiggan, J. C. Lester, Modeling and evaluating empathy in embodied companion agents, International Journal of Human-Computer Studies 65 (4) (2007) 348–360.

[77] T. Zhao, K. Lee, M. Eskenazi, Unsupervised discrete sentence representation learning for interpretable neural dialog generation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 1098–1107.

[78] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, E. P. Xing, Toward controlled generation of text, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 1587–1596.

[79] B. W. Schuller, A. M. Batliner, Emotion, affect and personality in speech and language processing.

[80] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A. Gelbukh, Sentiment and sarcasm classification with multitask learning, IEEE Intelligent Systems 34 (3) (2019) 38–43.

[81] J. Bang, H. Noh, Y. Kim, G. G. Lee, Example-based chat-oriented dialogue system with personalized long-term memory, in: 2015 International Conference on Big Data and Smart Computing (BIGCOMP), 2015, pp. 238–243.

[82] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, B. Dolan, A neural network approach to context-sensitive generation of conversational responses, pp. 196–205.

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

[84] R. Al-Rfou, M. Pickett, J. Snaider, Y. Sung, B. Strope, R. Kurzweil, Conversational contextual cues: The case of personalization and history for response ranking, CoRR.

[85] C. K. Joshi, F. Mi, B. Faltings, Personalization in goal-oriented dialog, arXiv preprint arXiv:1706.07503.

[86] K. Mo, Y. Zhang, S. Li, J. Li, Q. Yang, Personalizing a dialogue system with transfer reinforcement learning, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[87] Q. Qian, M. Huang, H. Zhao, J. Xu, X. Zhu, Assigning personality/profile to a chatting machine for coherent conversation generation, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 4279–4285.

[88] M. Yang, W. Tu, Q. Qu, Z. Zhao, X. Chen, J. Zhu, Personalized response generation by dual-learning based domain adaptation, Neural Networks 103 (2018) 72–82.

[89] Y. Zheng, G. Chen, M. Huang, S. Liu, X. Zhu, Personalized dialogue generation with diversified traits, arXiv preprint arXiv:1901.09672.

[90] Y. Zemlyanskiy, F. Sha, Aiming to know you better perhaps makes me a more engaging dialogue partner, in: Proceedings of the 22nd Conference on Computational Natural Language Learning, Association for Computational Linguistics, 2018, pp. 551–561.

[91] E. Chu, P. Vijayaraghavan, D. Roy, Learning perso nas from dialogue with attentive memory networks, in: Proceedings of the 2018 Conference

on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2018, pp. 2638–2646.

[92] O. Olabiyi, A. Khazane, E. T. Mueller, A persona-based multi-turn conversation model in an adversarial learning framework, in: 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, Orlando, FL, USA, December 17-20, 2018, 2018, pp. 489–494.

[93] L. Luo, W. Huang, Q. Zeng, Z. Nie, X. Sun, Learning personalized end-to-end goal-oriented dialog, arXiv preprint arXiv:1811.04604.

[94] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, B. Dolan, A persona-based neural conversation model, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 994–1003.

[95] A. Ritter, C. Cherry, W. B. Dolan, Data-driven response generation in social media, in: Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, 2011, pp. 583–593.

[96] R. Lowe, N. Pow, I. Serban, L. Charlin, J. Pineau, Incorporating unstructured textual knowledge sources into neural dialogue systems, in: Neural Information Processing Systems Workshop on Machine Learning for Spoken Language Understanding, 2015.

[97] M. Neff, Y. Wang, R. Abbott, M. Walker, Evaluating the effect of gesture and language on personality perception in conversational agents, in: J. Allbeck, N. Badler, T. Bickmore, C. Pelachaud, A. Safonova (Eds.), Intelligent Virtual Agents, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 222–235.

[98] F. Mairesse, M. A. Walker, Towards personality-based user adaptation: Psychologically informed stylistic language generation, User Modeling

and User-Adapted Interaction 20 (3) (2010) 227–278. doi:10.1007/s11257-010-9076-2.
URL https://doi.org/10.1007/s11257-010-9076-2

[99] F. Mairesse, M. A. Walker, Controlling user perceptions of linguistic style: Trainable generation of personality traits, Computational Linguistics 37 (3) (2011) 455–488.

[100] T. Polzehl, PERSONALITY IN SPEECH., Springer, 2016.

[101] K. Lee, W. Peng, S.-A. Jin, C. Yan, Can robots manifest personality?: An empirical test of personality recognition, social responses, and social presence in human–robot interaction, Journal of Communication 56 (2006) 754 – 772. doi:10.1111/j.1460-2466.2006.00318.x.

[102] C. Nass, K. M. Lee, Does computer-generated speech manifest personality? an experimental test of similarity-attraction, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '00, Association for Computing Machinery, New York, NY, USA, 2000, pp. 329–336. doi:10.1145/332040.332452.
URL https://doi.org/10.1145/332040.332452

[103] N. Howard, E. Cambria, Intention awareness: Improving upon situation awareness in human-centric environments, Human-centric Computing and Information Sciences 3 (9) (2013) .

[104] Y. Mehta, N. Majumder, A. Gelbukh, E. Cambria, Recent trends in deep learning based personality detection, Artificial Intelligence Review 53 (2020) 2313–2339.

[105] W. T. Norman, 2800 personality trait descriptors–normative operating characteristics for a university population.

[106] J. M. Digman, Personality structure: Emergence of the five-factor model, Annual review of psychology 41 (1) (1990) 417–440.

[107] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, J.-Y. Nie, A hierarchical recurrent encoder-decoder for generative context-aware query suggestion, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, ACM, New York, NY, USA, 2015, pp. 553–562.

[108] M. Gašić, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis, S. Young, POMDP-based dialogue manager adaptation to extended domains, in: Proceedings of the SIGDIAL 2013 Conference, Association for Computational Linguistics, Metz, France, 2013, pp. 214–222.

[109] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, S. Fidler, Skip-thought vectors, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS'15, MIT Press, Cambridge, MA, USA, 2015, pp. 3294–3302.

[110] S. Ji, S. Pan, E. Cambria, P. Marttinen, P. S. Yu, A survey on knowledge graphs: Representation, acquisition and applications, arXiv preprint arXiv:2002.00388.

[111] H. He, A. Balakrishnan, M. Eric, P. Liang, Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2017, pp. 1766–1776.

[112] A. Madotto, C.-S. Wu, P. Fung, Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2018, pp. 1468–1478.

[113] D. Guo, D. Tang, N. Duan, M. Zhou, J. Yin, Dialog-to-action: Conversational question answering over a large-scale knowledge base, in: Advances in Neural Information Processing Systems, 2018, pp. 2946–2955.

[114] C.-S. Wu, R. Socher, C. Xiong, Global-to-local memory pointer networks for task-oriented dialogue, arXiv preprint arXiv:1901.04713.

[115] L. Xu, Q. Zhou, K. Gong, X. Liang, J. Tang, L. Lin, End-to-end knowledge-routed relational dialogue system for automatic diagnosis, arXiv preprint arXiv:1901.10623.

[116] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-t. Yih, M. Galley, A knowledge-grounded neural conversation model, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[117] B. Liu, G. Tür, D. Hakkani-Tür, P. Shah, L. Heck, Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Vol. 1, 2018, pp. 2060–2069.

[118] R. Reddy, D. Contractor, D. Raghu, S. Joshi, Multi-level memory for task oriented dialogs, arXiv preprint arXiv:1810.10647.

[119] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, X. Zhu, Commonsense knowledge aware conversation generation with graph attention., in: IJCAI, 2018, pp. 4623–4629.

[120] Z. Wei, Q. Liu, B. Peng, H. Tou, T. Chen, X.-J. Huang, K.-F. Wong, X. Dai, Task-oriented dialogue system for automatic diagnosis, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2018, pp. 201–207.

[121] M. Eric, L. Krishnan, F. Charette, C. D. Manning, Key-value retrieval networks for task-oriented dialogue, in: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 2017, pp. 37–49.

[122] A. Bordes, Y.-L. Boureau, J. Weston, Learning end-to-end goal-oriented dialog, arXiv preprint arXiv:1605.07683.

[123] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, O. Tafjord, Think you have solved question answering? try arc, the ai2 reasoning challenge, arXiv preprint arXiv:1803.05457.

[124] S. Ostermann, M. Roth, A. Modi, S. Thater, M. Pinkal, Semeval-2018 task 11: Machine comprehension using commonsense knowledge, in: Proceedings of the 12th International Workshop on semantic evaluation, 2018, pp. 747–757.

[125] J. Su, K. Duh, X. Carreras (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016.

[126] L. E. Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, K. Suleman, Frames: A corpus for adding memory to goal-oriented dialogue systems, arXiv preprint arXiv:1704.00057.

[127] A. R. Neelakantan, Knowledge representation and reasoning with deep neural networks, Ph.D. thesis, University of Massachusetts Amherst (2017).

[128] A. Newell, J. C. Shaw, H. A. Simon, Report on a general problem solving program, in: IFIP congress, Vol. 256, Pittsburgh, PA, 1959, p. 64.

[129] A. Pease, G. Sutcliffe, N. Siegel, S. Trac, Large theory reasoning with sumo at casc, AI Communications 23 (2-3) (2010) 137–144.

[130] E. Cambria, A. Hussain, C. Havasi, C. Eckl, Common sense computing: From the society of mind to digital intuition and beyond, in: Biometric ID Management and Multimodal Communication, Vol. 5707 of Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2009, pp. 252–259.

[131] R. Speer, C. Havasi, Representing general relational knowledge in conceptnet 5., in: LREC, 2012, pp. 3679–3686.

[132] E. Cambria, S. Poria, D. Hazarika, K. Kwok, SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings, in: AAAI, 2018, pp. 1795–1802.

[133] E. Cambria, B. Schuller, Y. Xia, B. White, New avenues in knowledge bases for natural language processing, Knowledge-Based Systems 108 (2016) 1–4.

[134] X. Zhong, E. Cambria, A. Hussain, Extracting time expressions and named entities with constituent-based tagging schemes, Cognitive Computation (2020) .

[135] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, M. Huang, Augmenting end-to-end dialogue systems with commonsense knowledge, in: AAAI, 2018, pp. 4970–4977.

[136] O. Vinyals, M. Fortunato, N. Jaitly, Pointer networks, in: Advances in Neural Information Processing Systems, 2015, pp. 2692–2700.

[137] J. Gu, Z. Lu, H. Li, V. Li, Incorporating copying mechanism in sequence-to-sequence learning, in: Annual Meeting of the Association for Computational Linguistics (ACL), 2016, Association for Computational Linguistics., 2016.

[138] M. Eric, C. D. Manning, A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue, EACL 2017 (2017) 468.

[139] M. E. Pollack, Overloading intentions for efficient practical reasoning, Noûs 25 (4) (1991) 513–536.

[140] D. Tang, B. Qin, X. Feng, T. Liu, Effective lstms for target-dependent sentiment classification, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3298–3307.

[141] M. Zhang, Y. Zhang, D.-T. Vo, Gated neural networks for targeted sentiment analysis, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[142] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM, in: AAAI, 2018, pp. 5876–5883.

[143] D. Hazarika, S. Poria, A. Zadeh, E. Cambria, L.-P. Morency, R. Zimmermann, Conversational memory network for emotion recognition in dyadic dialogue videos, in: NAACL, 2018, pp. 2122–2132.

[144] T. Young, V. Pandelea, S. Poria, E. Cambria, Dialogue systems with audio context, Neurocomputing 388 (2020) 102–109.

[145] H. Chu, D. Li, S. Fidler, A face-to-face neural conversation model, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7113–7121.

[146] N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, E. Cambria, DialogueRNN: An attentive rnn for emotion detection in conversations, in: AAAI, 2019, pp. 6818–6825.

[147] I. Chaturvedi, R. Satapathy, S. Cavallari, E. Cambria, Fuzzy commonsense reasoning for multimodal sentiment analysis, Pattern Recognition Letters 125 (264-270).

[148] W. Zhao, H. Peng, S. Eger, E. Cambria, M. Yang, Towards scalable and reliable capsule networks for challenging NLP applications, in: ACL, 2019, pp. 1549–1559.

[149] R. Sukthanker, S. Poria, E. Cambria, R. Thirunavukarasu, Anaphora and coreference resolution: A review, Information Fusion 59 (2020) 139–162.

[150] R. Satapathy, E. Cambria, A. Nanetti, A. Hussain, A review of shorthand systems: From brachygraphy to microtext and beyond, Cognitive Computation (2020) .

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: