Artificial Intelligence Review (2024) 57:113
https://doi.org/10.1007/s10462-024-10744-z

Business chatbots with deep learning technologies: state-of-the-art, taxonomies, and future research directions

Yongxiang Zhang[1] · **Raymond Y. K. Lau**[1] · **Jingjun David Xu**[1] · **Yanghui Rao**[2] · **Yuefeng Li**[3]

Accepted: 6 March 2024 / Published online: 11 April 2024

Abstract
With the support of advanced hardware and software technology, Artificial Intelligence (AI) techniques, especially the increasing number of deep learning algorithms, have spawned the popularization of online intelligent services and accelerated the contemporary development and applications of chatbot systems. The promise of providing 24/7 uninterrupted business services and minimizing workforce costs has made business chatbots a hot topic due to the impact of the pandemic. It has attracted considerable attention from academic researchers and business practitioners. However, a thorough technical review of advanced chatbot technologies and their relevance and applications to various business domains is rare in the literature. The main contribution of this review article is the critical analysis of various chatbot development approaches and the underlying deep learning computational methods in the context of some business applications. We first conceptualize current business chatbot architectures and illustrate the technical characteristics of two common structures. Next, we explore the mainstream deep learning technologies in chatbot design from the perspective of computational methods and usages. Then, we propose a new framework to classify chatbot construction architectures and differentiate the traditional retrieval-based and generation-based chatbots in terms of the modern pipeline and end-to-end structures. Finally, we highlight future research directions for business chatbots to enable researchers to devote their efforts to the most promising research topics and commercial scenarios and for practitioners to benefit from realizing the trend in business chatbot development and applications.

Keywords Business chatbot · Dialogue system · Deep learning · Reinforcement learning · Artificial intelligence

1 Introduction

The rapid development of the Internet and mobile intelligent devices has motivated more enterprises to develop their online business while creating new demand for developing user-friendly and effective human–computer interaction technologies such as chatbots. A business chatbot refers to a conversational agent that can interact with users via text

Extended author information available on the last page of the article

(Przegalinska et al. 2019), image (Chiu and Chuang 2021), or voice (Sanchez-Diaz et al. 2018) to accomplish specific commercial tasks, such as customer relationship management (Steinbauer et al. 2019), financial investment advising (Wen 2018), online shopping (Thomas 2016) and so on. During the pandemic, the global demand for epidemic prevention resulted in increased online business demand and higher requirements for 24/7 services. Using chatbot technology to automate messaging services with customers could reduce customer servicing costs or save on the operating costs of entire call centers (Lee et al. 2018). Considering the rapid increase in chatbot demand to improve customer services, a thorough study of state-of-the-art technology in business chatbots can help popularize chatbot applications and identify potential business values from human–computer interactions (Steinhoff et al. 2019). It is beneficial to chatbot developers, researchers, end-users, and sellers.

Various computational methods from multiple disciplines have been applied to chatbot design. The evolution of its mainstream techniques follows a similar development pattern as that of information technology (IT), including the algorithm, computer language, operating system, hardware, and Internet (Lokman and Zain 2010). Nowadays, chatbot development is in the middle of a boom driven by contemporary AI technologies. In particular, recent deep learning advancements have led to various innovative and effective approaches to improving and enriching chatbot functionalities. A qualified business chatbot is expected to understand the context and user intent, integrate with domain knowledge (Luo et al. 2021), orchestrate workflows within a customer relationship management (CRM) system (Jonke and Volkwein 2018), deliver personalized experiences and emotional values (Zhu et al. 2022), and collaborate with humans to ensure efficient and secure services (Ferrod et al. 2021; Yang et al. 2021b).

In comparison, chatbots in other fields may not require all these characteristics. For example, chatbots designed for psychological counseling may not need to integrate with CRM systems because they only interact with individuals and record their mental health status. Educational chatbots responsible for student assignments are not required to provide emotional values. However, to our knowledge, a literature review outlining the manifold deep learning technologies and identifying their utilization has been lacking in business chatbot studies. Conducting a review that presents the technology preferences of researchers in the business chatbot literature will be helpful because it can provide focused insights for further business value exploration. Through a review, a better understanding of researcher-preferred techniques and recognition of the technical and application gaps can be achieved. Hence, such a review is expected to help researchers and developers delve into the value of chatbots under specific commercial applications, considering the distinct characteristics and applicable scenarios related to diverse deep learning branches. Accordingly, we focus on the thorough review and analysis of mainstream deep learning technologies in the dialogue system, which produces the core service of business chatbots. It is an essential component that can significantly enrich user experience during human–computer interactions.

We start this systematic literature review following the guidelines proposed by Petersen et al. (2015), which were upgraded from Petersen et al. (2008). According to their summary, literature review has two types: systematic review and mapping study. Our literature review belongs to the review paradigm of mapping study, which aims at structuring a research area through classifying and counting contributions concerning the categories of that classification (Petersen et al. 2008). To provide differentiated insight into this burgeoning technique and commercial use, we aim to contribute in the following aspects: (1) provide conclusions on the current development structure of chatbots and recognize the

Table 1 Definition of search strings

Topic terms	Search keywords
Chatbot	chatbot* OR chatterbot* OR "dialog* system*" OR "conversation* system*" OR "conversation* agent*" OR "intelligent agent*" OR "virtual agent*" OR "intelligent assistant*" OR "virtual assistant*"
Business	business OR commerce OR customer OR consumer OR industry OR trade OR transaction
Deep learning	'deep learning' OR 'neural network*'

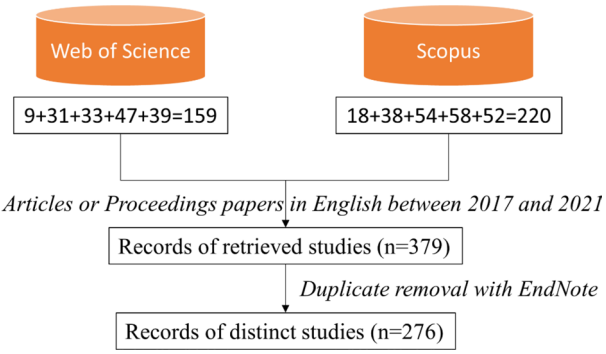

Fig. 1 Process of filtering literature from WoC and Scopus

mainstream deep learning techniques adopted for contemporary business chatbot building; (2) summarize the usages of the deep learning technologies and compare their performance in each usage; and (3) propose a comprehensive classification framework to characterize different chatbot architectures in terms of the response-producing ways.

Hence, we identify three sets of keywords from the research motivation: chatbot, business, and deep learning. Each set of searches is performed on the databases of Web of Science (WoC) and Scopus using a Boolean expression ("OR," "AND"). The search strings used for each database can be found in Table 1 and have been applied to all fields. This review paper focuses on the publications of English proceedings papers or articles in the past 5 years (2017 to 2021) to ensure the timeliness of recognized technologies. The process of filtering literature from two databases is shown in Fig. 1, and the distribution of the search results is shown in Fig. 2.

To outline our mapping study, we created a taxonomy diagram after reviewing all the qualified literature, as shown in Fig. 3. Deep learning technologies have four usages (in blue boxes) in chatbot dialogue systems, and some usages can be subdivided further (in purple boxes). We identify the mainstream deep learning techniques (in orange boxes) mentioned in the literature for each use and provide relevant references (in green boxes). We will conduct a more detailed study surrounding this taxonomy in Sects. 3 and 4.

The remainder of this article is organized as follows. Section 2 introduces the chatbot background and emphasizes the pipeline and end-to-end development structures at the current stage. Section 3 summarizes seven deep learning technologies from business chatbot literature and describes their computational mechanisms in dialogue systems. Section 4 presents the main applications of deep learning and compares various neural network characteristics for the same usage, followed by Sect. 5, which conducts a critical analysis of technical features for four chatbot architectures. Sections 6 and 7 discuss the future research directions and conclude the review article, respectively.

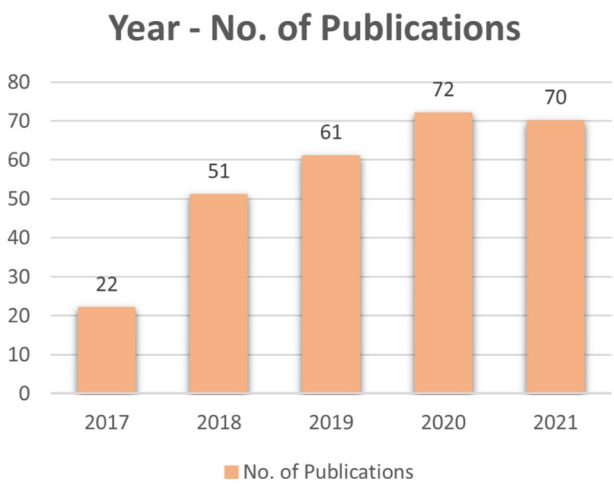

Fig. 2 Numbers of retrieved publications over time (2017–2021)

2 Background

Generally, a chatbot is composed of three parts: (1) a user interface for receiving inputs and delivering outputs; (2) the management system to provide a variety of chatbot core services; and (3) hardware support for the entire operation of a chatbot system. The general system architecture of a chatbot is shown in Fig. 4.

The interface of a chatbot receives an input message or instruction from a user and transmits it to the respective service management components. The service management components accept the input request and assign tasks to the respective sub-service components for operations. If a chatbot receives a query, the dialogue management system needs to produce an appropriate response to start a conversation with the user. The functionalities of a chatbot can be enriched by embedding it with internal or external application programming interfaces (APIs), such as the APIs for e-mail service, external messaging, geographical maps, weather, and other standard utility services. The chatbot interface also receives responses from the service management system and formats the responses according to the specific presentation style required by the user. These chatbot services are executed by invoking hardware components, including temporary storage for dialogue and service data, permanent memory for fundamental system operations, central processing units and graphical processors, intelligent routers for Internet connections, and so on. The automatic reply (auto-responder) of the dialogue system is one of the most primitive functions of chatbots (Mufadhol et al. 2020), and its benefit of emancipating productivity is one of the well-recognized values of developing business chatbots (Sandu and Gide 2019; Steinbauer et al. 2019).

Starting in the early twenty-first century, a surge in machine learning and deep learning research has spread to chatbot studies (Adamopoulou and Moussiades 2020), advancing chatbots gradually into becoming more intelligent and modernized. The research and development of business chatbots have entered an era of explosive growth. Contemporary chatbots are designed to provide context-sensitive responses and deliver an array of sophisticated functionalities. They have gradually evolved into two common architectures: the pipeline and end-to-end structure. Figure 5 illustrates the information processing flow of a dialogue system in the pipeline structure. Conceptual components in this kind of

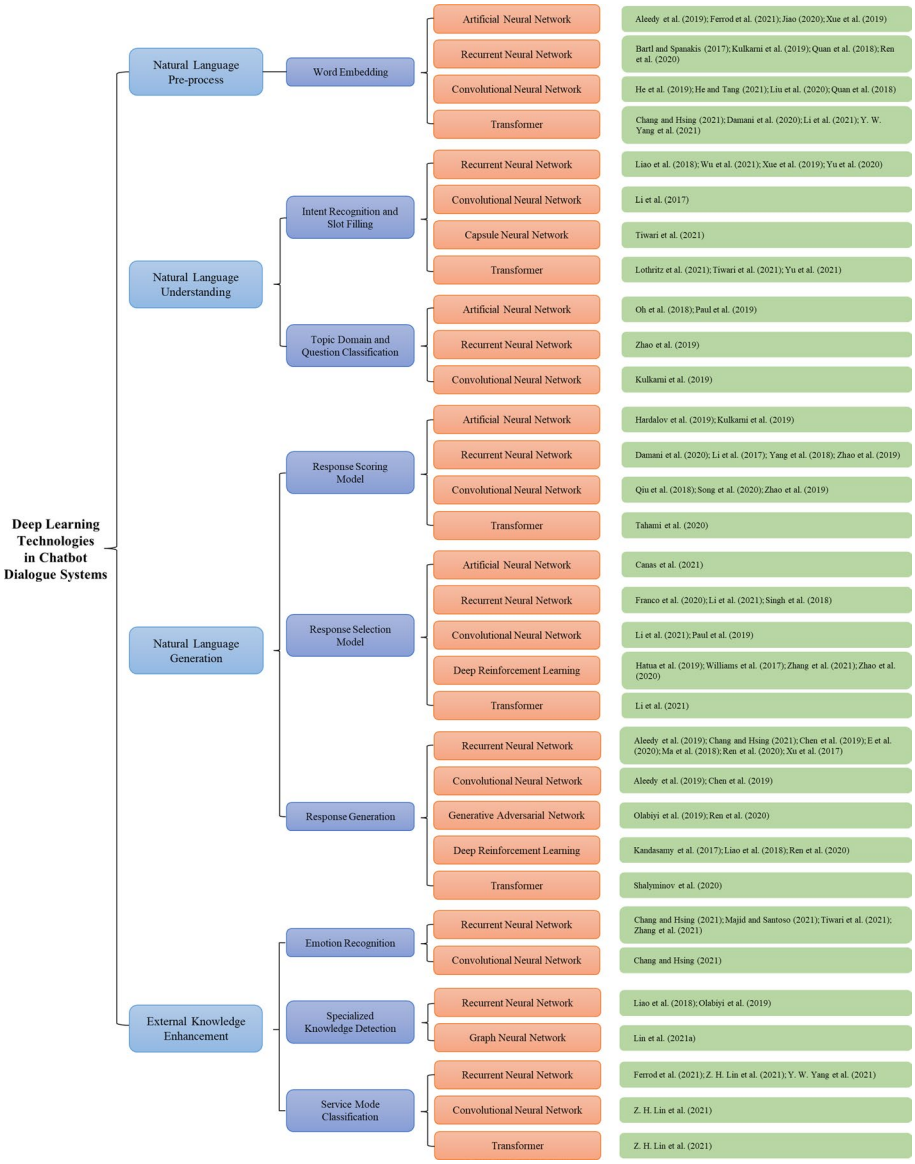

Fig. 3 Taxonomy of deep learning usages in chatbot dialogue systems

conversation system generally include natural language understanding (NLU) for dialogue intent and slot recognition, dialogue state tracking (DST) for conversation record management, dialogue policy learning (DPL) for response policy controlling, and natural language generation (NLG) for response generation (Chen et al. 2018; Hirschberg and Manning 2015). In some cases, the DST and DPL components will be integrated into a united concept named dialogue management (DM), which is responsible for storing and controlling the conversation states. Current studies might weaken the DM conception and distribute its

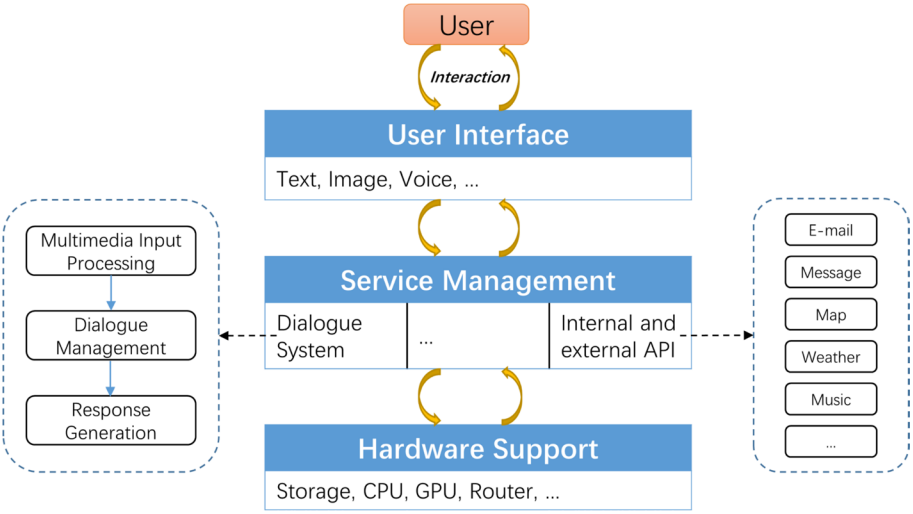

Fig. 4 General architecture of a chatbot system

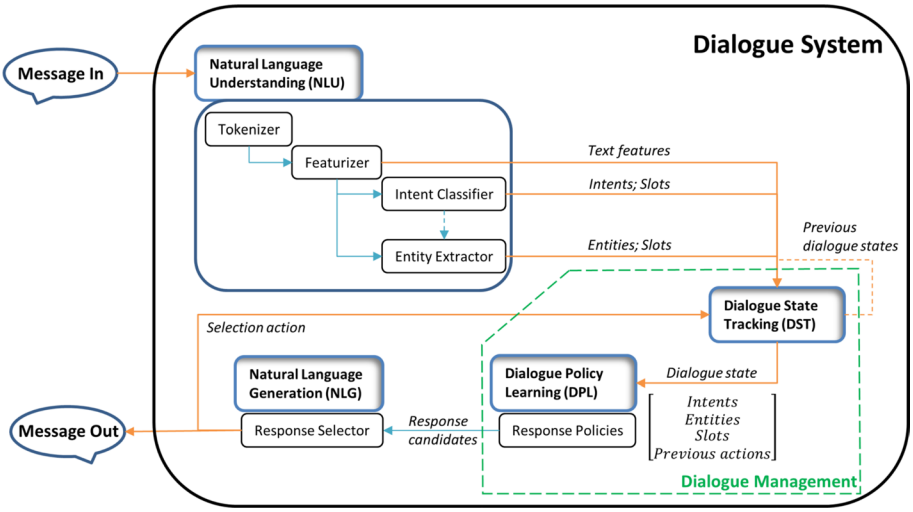

Fig. 5 Illustration of a dialogue system in the pipeline structure

functionality to NLU or NLG components when not emphasizing the dialogue state management in system design. Chatbots with this architecture explicitly present the response generation process to developers and researchers for the convenience of frequently adjusting or independently improving each component with user requirements adapted.

Building a chatbot in pipeline architecture requires massive manual operations for designers, especially in adapting the NLU module to specific application scenarios. To lessen manual intervention, some researchers prefer the other kind of architecture, the end-to-end structure. An example is shown in Fig. 6; the dialogue system operates with raw data input and directly outputs the final processed results. Except for the text vectorization

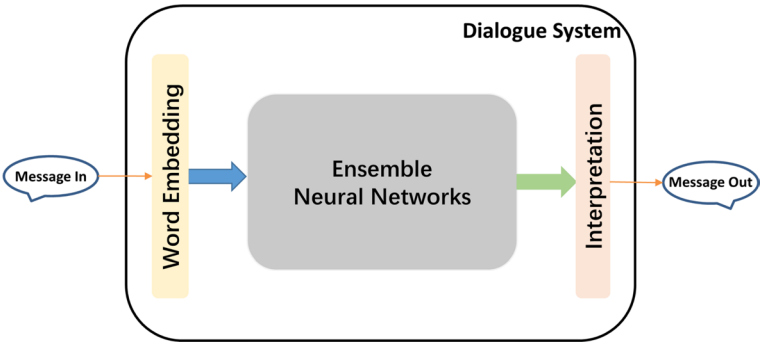

Fig. 6 Illustration of a dialogue system in the end-to-end structure

and result interpretation, the remaining computational space allows the ensemble neural network to adjust its model parameters automatically according to the training data. This structure increases the model's overall fit but requires massive data to "teach" the model to learn the intrinsic data relationships (Yang et al. 2019). The choice of two structures depends on the actual business needs and designers' preferences.

To facilitate both academic research and practical application, several reputable enterprises have provided their respective open-source development frameworks that can unify and simplify chatbot design, including Google's DialogFlow, Facebook's Wit.ai and Messenger, and Microsoft Bot Framework, Amazon Alexa, IBM Watson Assistant, and Rasa's RASA. Some tools may provide development structures to satisfy designers' diverse needs. For example, RASA is a machine learning infrastructure to automate text-based conversations, providing pipeline and end-to-end developing libraries (Bocklisch et al. 2017). In the pipeline structure, each dialogue system component is trained and can be replaced by equivalent methods independently. In contrast, the ensemble model in end-to-end design is trained simultaneously, where the parameter updates of each subpart affect each other. These two architectures have developed distinctive construction advantages to shape the mainstream options. A detailed comparative analysis involving architecture characteristics will be carried out along with their internal deep learning applications in Sect. 5.

3 Mainstream deep learning methods for business chatbot development

Deep learning refers to a series of burgeoning computational technologies with a unique operation structure named neural network (LeCun et al. 2015), originating from the biological neural networks that constitute animal brains (McCulloch and Pitts 2016). A computational deep learning model comprises many nodes (or artificial neurons) connecting in diverse forms. Different connection and operation ways constitute various artificial neural networks. Although Rosenblatt (1958) created the first neural network and Rumelhart et al. (1986) designed the backpropagation algorithm to train the model decades ago, deep learning research was stuck for a long time due to the limitation of computing capability. Driven by the rapid development of modern computer software and hardware technology, the feasibility and potential of deep learning have revived researchers' interest in diverse neural network algorithms. They enrich the machine learning family and expand AI influence

with its ground-breaking capability of multidimensional data processing. In this section, we identify several mainstream deep learning technologies from recent business chatbot research and briefly introduce their characteristics and computational mechanisms. A concise summary of the technologies and their characteristics is shown in Table 2.

3.1 Artificial neural network

Artificial neural network (ANN), also called the neural network, is a mathematical model that imitates the structure and function of a biological neural network to estimate or approximate the nonlinear functional relationship between the network inputs and outputs. An ANN comprises many nodes connected in different ways to convey information. Each non-input node represents a specific output function called the activation function. Each connection (edge) between two nodes represents a weighted value for the signal passing through the connection, equivalent to the ANN memory. We usually divide the neural network into the input, output, and hidden layers according to the input and output positions of the model. The network output varies with the connection mode (network structure), weight values of edges, and activation functions of nodes for accomplishing different tasks.

An instance of a simple neural network is illustrated in Fig. 7. It is a fully connected feed-forward neural network where each neuron in a layer connects to each node in the adjacent layer. Its information flows only in a forward direction (from the input to the hidden to the output layers) without a cycle or loop connection (Schmidhuber 2015). This sample model has one hidden layer with N dimensions, and the operations of the different layers are as follows.

$$\boldsymbol{h} = \sigma_h(\boldsymbol{W_h x} + b_h)$$

$$y = \sigma_y(\boldsymbol{W_y h} + b_y)$$

where $\boldsymbol{x} = \{x_0, x_1, \ldots, x_i, \ldots, x_{M-1}\} i \in (0, M)$ stands for an M-dimensional input vector, $\boldsymbol{h} = \{h_0, h_1, \ldots, h_j, \ldots, h_{N-1}\} j \in (0, N)$ and y stand for the hidden layer vector and output of the network, respectively, $\boldsymbol{W_h}$ and $\boldsymbol{W_y}$ are the weight matrices, b_h and b_y are the bias terms in the corresponding layer, and σ_h and σ_y stand for the activation functions that are often Rectified Linear Unit (ReLU), tanh, or sigmoid to calculate a weighted sum of the inputs in the node.

An ANN with few hidden layers is named a shallow neural network and has been explored for decades in the last century (Schmidhuber 2015). In 2006, Hinton et al. (2006) alleviated the local optimal solution problem using pre-training methods using an ANN with seven hidden layers, allowing for deep-layer operations and rekindling people's attention to deep learning. An ANN with complex neural structures and many network layers is called a Deep Neural Network (DNN). For example, a fully connected neural network with multiple hidden layers is a standard DNN. DNN has dominated ANN applications due to its robust feature extraction and learning ability. The following sections introduce several common DNN variants in business chatbots.

Table 2 Summary of mainstream deep learning technologies and their uniqueness

Technology	Uniqueness	Classic Literature
Fully connected Feed-forward neural network	Each neuron in one layer connects to each node in the adjacent layer; the information flows only in a forward direction without a cycle or loop connection	Rosenblatt (1958), Rumelhart et al. (1986), LeCun et al. (2015) and Schmidhuber (2015)
Recurrent neural network	It has directed cycles in model memory that can allow the temporal sequence as input and is specialized in textual data processing to learn the semantic relationships between words	Jordan (1986), Elman (1990), Hochreiter and Schmidhuber (1997), Cho et al. (2014), Sutskever et al. (2014), Zhou et al. (2016) and Serban et al. (2016)
Convolutional neural network	It replaces the general matrix multiplication with the convolution operation in at least one network layer; the sequential operations in the convolutional layer and pooling layer enable the network to cope with two and three-dimensional data	Fukushima (1980), Weng et al. (1993), Kim (2014), Zhang and Wallace (2015) and Goodfellow et al. (2016)
Capsule neural network	It utilizes a vector as the model output to represent the spatial information and probability value of a detected pattern; it overcomes the shortcomings of the max-pooling operations that would cause valuable spatial features to be lost	Sabour et al. (2017) and Zhao et al. (2018)
Graph neural network	It has a "graph-in, graph-out" architecture and transforms the embeddings of nodes and edges without changing the connectivity of the input graph	Gori et al. (2005), Scarselli et al. (2009) and Zhou et al. (2020)
Generative adversarial network	It consists of two sub-models: a generative model to approximate the data distribution and a discriminative model to estimate the probability that a sample came from the real data rather than the generative model	Goodfellow et al. (2014), Kusner and Hernández-Lobato (2016), Zhang et al. (2016), Zhang et al. (2017), Yu et al. (2017) and Guo et al. (2018)
Deep reinforcement learning	It combines reinforcement learning with deep learning to optimize the objective function and make better decisions in the sequential decision problem	Otterlo and Wiering (2012), Mnih et al. (2015), Silver et al. (2016) and Serban et al. (2017)
Transformer	It fully uses the self-attention mechanism in the encoder-decoder structure that allows reading an entire sentence simultaneously	Vaswani et al. (2017), Radford et al. (2018) and Devlin et al. (2019)

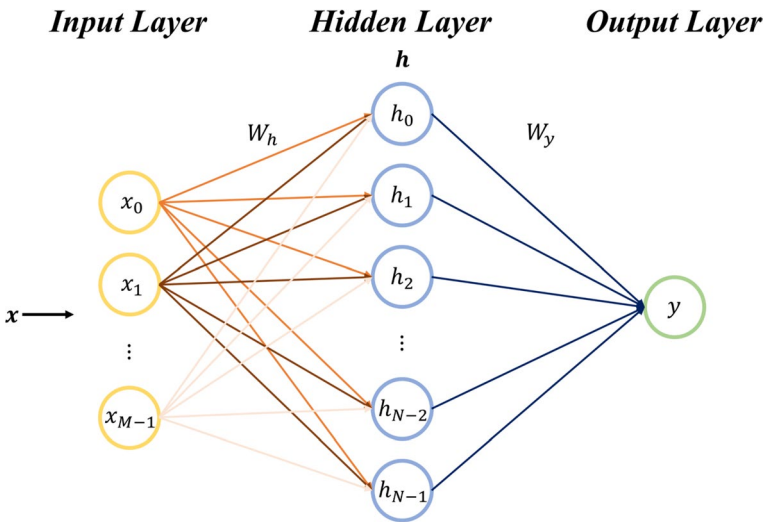

Fig. 7 Double-layer fully connected feed-forward neural network

3.2 Recurrent neural network

A recurrent neural network (RNN) is an essential branch of deep learning technologies with directed cycles in model memory that can allow the temporal sequence as input. It is particularly specialized in textual data processing to capture the semantic relationship between words, with the information stored in the multidimensional weights of the networks. The rise of chatbots that can "generate" a response also greatly benefits from RNN development. Two primitive types of RNN have the characteristic of using the internal state (memory) stored in the hidden layer unit to process sequences of inputs. Jordan network (Jordan 1986) uses the output of the output layer at a previous time point as one of the inputs in the current hidden layer. Elman network (Elman 1990), a more general RNN, uses the output of the hidden layer at a previous time point as one of the inputs in the current hidden layer. The inputs of the hidden layer distinguish these two networks.

However, simple RNN models often have difficulty obtaining satisfactory results. The gradient vanishing will happen with the gradient exploding in the simple RNN model, which is difficult to solve by adjusting the learning rate or other model parameters. LSTM is an efficient gradient-based method Hochreiter and Schmidhuber (1997) proposed to alleviate the gradient vanishing problem. Compared to simple RNN units, LSTM adds three logic gates (input, forget, and output) to control the input information, long-term memory, and short-term memory. These gates enable the model to solve the gradient vanishing problem so that the learning rate can be set small. GRU (Cho et al. 2014) is another RNN variant similar to LSTM, and its most prominent characteristic is reducing three gates to two (update and reset); hence, the training speed is accelerated. The GRU model can only read in the new input if it empties the existing state. The performances of these two structures are not far apart but are better than those of traditional recurrent units (Chung et al. 2014).

With the efforts of deep learning researchers, RNN has formed various typical structures after multiple iterations. Seq2seq is one of the most groundbreaking and far-reaching designs. It is a general end-to-end model in an encoder-decoder structure proposed by Cho

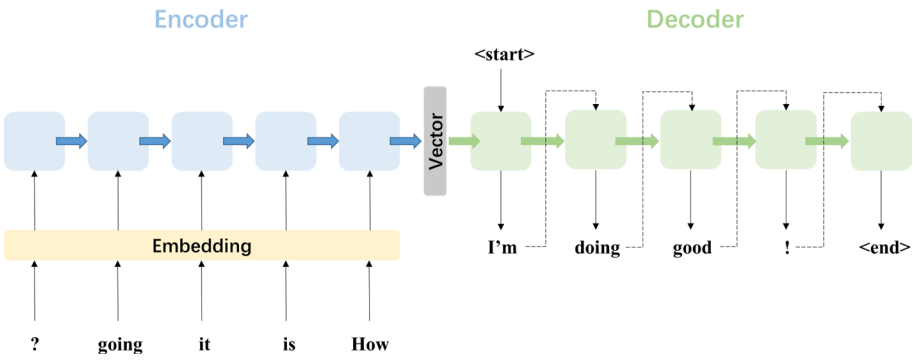

Fig. 8 Seq2seq model

et al. (2014) and Sutskever et al. (2014). As shown in Fig. 8, it comprises two RNNs, an encoder and a decoder. One RNN encodes a sequence of word vectors into a fixed-length vector representation, and the other decodes the representation into another sequence. The units in this model (blue and green squares in Fig. 8) are usually LSTM or GRU. The encoder and decoder are trained jointly to maximize the conditional probability of a target sequence based on the given source sequence (Cho et al. 2014). This kind of design that can generate a response word by word has been widely used in text generation tasks.

Other well-known improvements include the Bi-LSTM design and hierarchical recurrent encoder-decoder (HRED) architecture. Zhou et al. (2016) proposed the Bi-LSTM that could read the input sequence from two directions to utilize the contextual features instead of the one-side information at a specific time. HRED was proposed by Serban et al. (2016) to model the contextual information for dialogue generation. They considered the turn-taking nature of dialogues and added a context encoder based on the Seq2seq structure to encode the temporal features of appeared utterances. This contextual modeling can reduce the computational steps among adjacent sentences and facilitate the propagation of the information and gradients in the network, thus enabling multi-turn conversation.

3.3 Convolutional neural network

A convolutional neural network (CNN) is a special neural network that replaces the general matrix multiplication with the convolution operation in at least one network layer (Goodfellow et al. 2016). Inspired by the research findings of the cat and monkey's visual cortices (Hubel and Wiesel 1959, 1968), Fukushima (1980) proposed a neural network composed of a convolutional and a downsampling layer, and the max-pooling computation approach was further introduced by Weng et al. (1993) for downsampling. A max-pooling layer often follows a convolution layer in modern CNN design. The uniqueness of CNN architecture lies in extracting spatial features through sequential operations in the convolutional and pooling layers, which validly enables the network to cope with two and three-dimensional input data.

Kim (2014) and Zhang and Wallace (2015) developed the TextCNN for sentence classification tasks to adapt the CNN model for text analysis. A simplified illustration is shown in Fig. 9. Each word in the input sentence will be transformed into a vector with word embedding techniques, and the combined matrix represents the whole sentence. Suppose

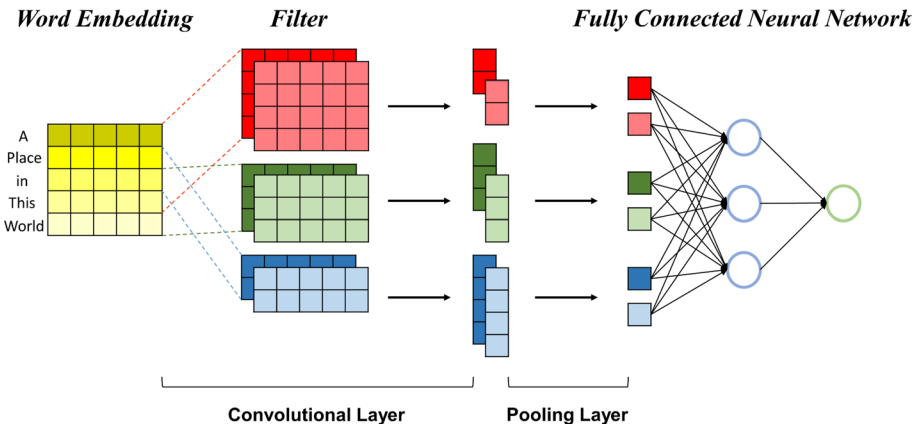

Fig. 9 CNN's computational mechanism for textual data

an input sentence comprises N word vectors of D dimensions, an $N \times D$ word matrix. The CNN model in Fig. 9 has six filters of three window sizes, each corresponding to two filters with different values to capture distinct information. The window size (note as M) represents the number of word vectors an $M \times D$ filter can cover in a convolution operation. Each filter slides along the input matrix to perform convolution operations in the convolutional layer, and the computed $(N - M + 1) \times 1$ vector is passed to the next pooling layer. A pooling operation is essentially a nonlinear form of downsampling that can reduce the dimensions of the received vector. The maximum value will be extracted from the vector for the frequently used max-pooling. In this instance, six values produced by the pooling layer correspond to the convolution and pooling results of six filters; they will be conveyed to a fully connected network to carry out the final calculation.

3.4 Capsule neural network

The capsule neural network (CapsNet) was initially proposed by Sabour et al. (2017) to overcome CNN's shortcoming that the max-pooling layer could only concentrate on detecting important local features in computer vision applications. Due to the lack of relative positions among the features, CNN models would misidentify an image of a face with disordered features as correct or an oblique real face as wrong in cases where sufficient data samples are unavailable. Hence, Sabour et al. (2017) invented a special neuron called a capsule to replace the max-pooling operation they thought was causing valuable spatial features to be lost. The most distinguishing characteristic of the capsule neuron is it utilizes a vector as the model output to represent the spatial information (the direction of a vector) and probability value (the norm of a vector) of a detected pattern. This design enables the network to intelligently model the intrinsic spatial relationship between a part and a whole, automatically generalizing the learned knowledge to novel viewpoints and reducing the sample size requirement. Similar to CNN, CapsNet has also been applied to textual data mining; it was first adapted by Zhao et al. (2018) for text classification tasks.

The input of a CapsNet model is usually multiple vectors. Figure 10 illustrates a capsule neuron with two input vectors reshaped from the computed results of the convolutional layer in Fig. 9. The affine transformation is first applied to the input vectors ($\boldsymbol{v}_1$ and

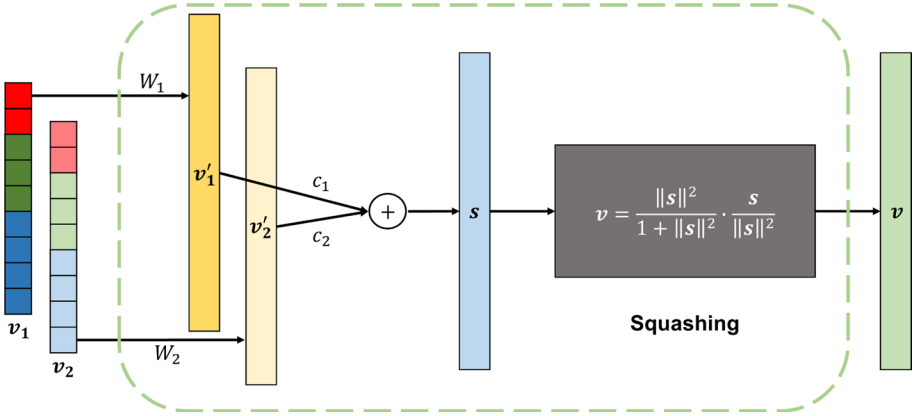

Fig. 10 Computational mechanism of a capsule neuron

v_2) to convert the low-dimensional feature vectors into high-dimensional ones (v'_1 and v'_2). Transformed vectors are then computed by weighted sum to calculate a new vector s. The weights c_1 and c_2 (named coupling coefficients) are determined by the dynamic routing algorithm (Sabour et al. 2017), and the sum of the coupling coefficients equals 1. Finally, vector s is processed by a "squashing" operation (the formula is shown in the grey box of Fig. 10) to generate the output vector v with the norm between 0 and 1. This value represents the predicted probability of the pattern that the neuron is responsible for detecting.

3.5 Graph neural network

A graph neural network (GNN) is an optimizable transformation on all attributes of a graph (nodes, edges, global-context) that preserves permutation equivariant architecture (Sanchez-Lengeling et al. 2021). The concept was proposed by Gori et al. (2005) and further elaborated by Scarselli et al. (2009). GNN adopts a "graph-in, graph-out" architecture, as shown in Fig. 11. Its input is a graph with feature information loaded into its nodes, edges, and global context, and the model progressively transforms these embeddings without changing the connectivity of the input graph.

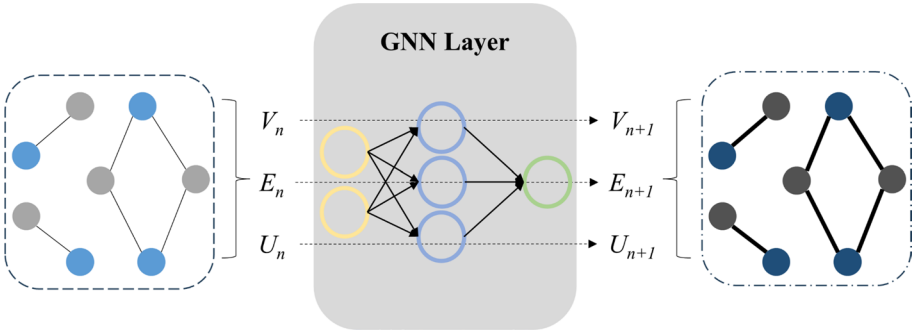

Fig. 11 One GNN layer for graphic data

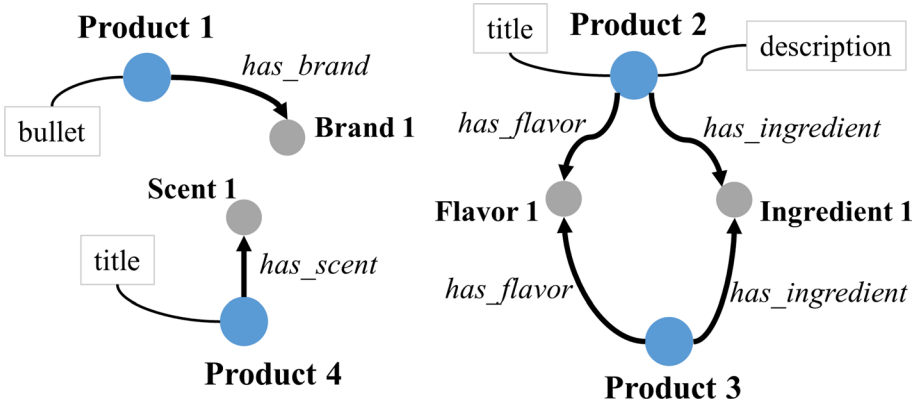

Fig. 12 Product knowledge graph used in the beauty shopping contexts

In a business chatbot, GNN is usually associated with a knowledge graph, a semantic network that can provide additional domain knowledge and assist chatbots in downstream decision-making. A knowledge graph represents a network of entities (nodes) and illustrates their relationships (edges). The entity can be any object, event, situation, or abstract concept. Each node has its attributes. For example, Fig. 12 presents a product knowledge graph Lin et al. (2021a) used for beauty shops. Take the node *Product 2* and its relations as an example. It connects two nodes, the entities *Ingredient 1* and *Flavor 1*, through the relations *has_ingredient* and *has_flavor*, respectively. Its attributes include *title* and *description* in this knowledge graph.

GNN is adopted to process such a graph structure as input and produce new representations embedded with graph information for nodes or edges. Usually, the graph can be modeled as $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the node set and $\mathbf{E}$ is the edge set. $\mathbf{U}$ is the global-context set, such as the number of nodes and edges. The corresponding feature representations of the nodes, edges, and global contexts in the GNN's n-th layer are V_n, E_n, and U_n. The product knowledge graph in Fig. 12 illustrates the entity attribute information, and thus, the node feature matrix V_n can be considered here. For node i, the GNN's n-th layer updates its feature representation from $V_{i.n}$ to $V_{i.n+1}$, aggregating information from its attributes and immediate neighbors with a specific optimizable transformation method. Zhou et al. (2020) have summarized existing optimizable transformation methods, mainly including the convolution operator, recurrent operator, skip connection, etc. The updated feature representations of all nodes V_{n+1} will be passed to the next layer $n+1$, and outputs of the GNN's final layer are new node representations that can be further used for downstream tasks. After GNN's iterative processing, the representations for nodes have integrated with the whole graph information.

3.6 Generative adversarial network

The generative adversarial network (GAN) is a fascinating semi-supervised learning framework that Goodfellow et al. (2014) proposed for estimating generative models. It is widely used in computer vision to generate new image samples initially. It comprises two sub-models: a generative model (generator) to approximate the data distribution and a discriminative model (discriminator) to estimate the probability that a sample came from the

real data rather than the generative model. These two parts are trained simultaneously in the form of a minimax two-player game, and the training procedure for the generator is to maximize the probability of the discriminator making a mistake.

Seq2seq-based artifacts are the most common generative model for text generation. However, they suffer from a severe problem in that the generative model tends to produce a safe response (Zhang et al. 2018b) without practical significance, such as "I don't know" or "I think so." The main reason comes from the effects of many safe answers in the training corpus, with many responses starting with "I." The probability distribution of words in different sentence positions has an apparent long tail characteristic. Hence, the decoder would be affected to select the most probable "I" as the first word of the response, further affecting the generation of subsequent words. The appearance of a safe response implies that the Seq2seq model is trapped in a locally optimal solution. A feasible operation is to impose a disturbance, such as GAN, on the model to make it jump out of the local solution and enter a more optimal global state, thus alleviating the problem.

Figure 13 briefly illustrates a response generation model implemented in a GAN schema with the Seq2seq model as the generator and another neural network as the discriminator. The generator outputs a generated utterance based on the received input that might be processed in some noises or interferences (if any), such as masking some words of the input utterance (Fedus et al. 2018), to increase the conversation variety. The discriminator is responsible for judging the gap between the generated text and the real one and transmitting the computed loss to update the generator parameters. The upgrade of this powerful framework owes much to the work of some researchers who contributed to migrating the GAN mechanism from image creation to text generation. Kusner and Hernández-Lobato (2016) introduced the Gumbel-softmax distribution for the generator to handle the discrete sequence data generation problem. Zhang et al. (2016) designed the textGAN model with a feature distribution matching method to solve a similar text generation problem, and they further improved the objective function with a kernelized discrepancy metric to ameliorate the mode-collapsing problem in GAN's training process (Zhang et al. 2017). In addition, Yu et al. (2017) and Guo et al. (2018) proposed the SeqGAN and LeakGAN models to optimize the generative performance of long text

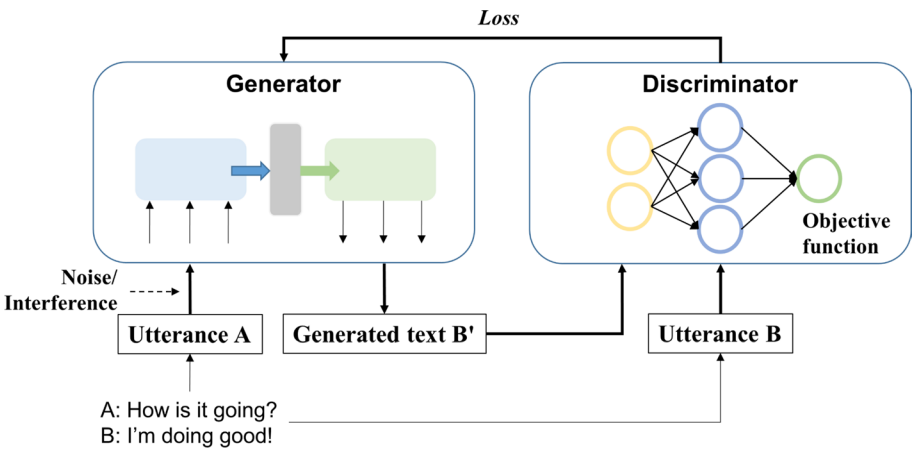

Fig. 13 GAN schema for text generation

with policy gradient-based reinforcement learning. These GAN-based models have a prominent characteristic of generating from text to text with a similar structure, which brings the models the potential to solve data paucity problems. They can enhance the original data by this kind of text retelling and improve the robustness of unknown data by data augmentation from the generative model.

3.7 Deep reinforcement learning

Deep reinforcement learning (DRL) refers to a series of algorithms that blend deep learning with reinforcement learning to optimize objective functions and make better decisions in sequential decision problems. Reinforcement learning is a cyclic process in which an agent takes actions according to explicit or implicit policies and then interacts with the environment to gain a reward and change its perceived state. It is designed to maximize the cumulative reward (formulated as the objective function) and settle the decision optimization problem. The distinct characteristic is to consider long-term income through frequent sequential interactions in a trial-and-error mechanism.

The reinforcement learning environment was initially abstracted as a Markov Decision Process (MDP) and solved with the dynamic programming method (Otterlo and Wiering 2012; Bellman 1954). It required much computation time and space to figure out the state transition process and caused the algorithm development to stagnate. The renaissance primarily comes from an event where a computer Go program, AlphaGo, won the Go world champion Fan Hui without handicap on a full-sized 19×19 board in 2015. AlphaGo was deployed with the DRL that united function approximation and objective optimization. It leveraged the deep learning perceptual ability to retain the state transition and modeled the policy and objective function of reinforcement learning. The program leader, David Sliver, was thus written into history as the pioneer of DRL, and the achievement has been published in *Nature* (Mnih et al. 2015; Silver et al. 2016).

A typical DRL process is shown in Fig. 14. Regarding the environment observed at time step t, the agent constructs a perceived state S_t and follows a value-based or a policy-based method to map from the state to an action A_t. Then, the environment reacts to the agent's action with a scalar reward R_t. The agent receives feedback from the environment

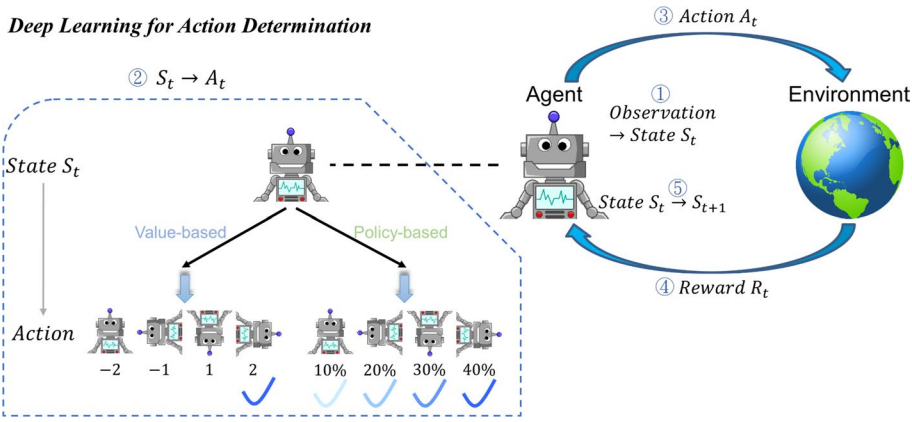

Fig. 14 Overview of the DRL process

and updates its perception into a new state S_{t+1}. Deep learning is leveraged to capture complicated relationships among the state, the agent action, and the reward or objective function for action determination and policy learning.

The agent built with a value-based method can utilize deep learning models to simulate a value function that estimates the cumulative reward. It selects the response action of the highest value, and one classical DRL algorithm is Deep Q-Network (DQN) developed by DeepMind (Mnih et al. 2015). The policy-based methods, such as Policy Gradients (Sutton et al. 2000), compute a probability distribution over actions according to the learned policy (represented by network parameters), so every action is likely to be chosen and is especially applicable to the scenario with a continuous action space. To sum up, DRL combines the perceptual ability of deep learning and the decision-making ability of reinforcement learning to define a decision problem and optimize the objective function in consideration of long-term benefits. Serban et al. (2017) adopted it in a social bot design with the dialogue process as state and the candidate response as action to improve users' multi-round interactive experience. Their Amazon Alexa Prize winning highlights DRL's feasibility in conversation systems.

3.8 Transformer

Transformer is an astonishing deep learning architecture initially proposed by a team from Google Brain (Vaswani et al. 2017) to accomplish machine translation tasks. It has been extended to various AI applications and is widely used to handle sequential data. Concerning traditional deep learning technologies for processing sequence data, the convolutional operation makes CNN essentially a type of local encoding of n-gram models, limiting the network's capability to establish long-range dependencies. And though the recurrent structure of RNN can capture long sequence information, it is challenging to run in parallel, often costing a tremendous amount of time. To process an entire sentence parallelly and

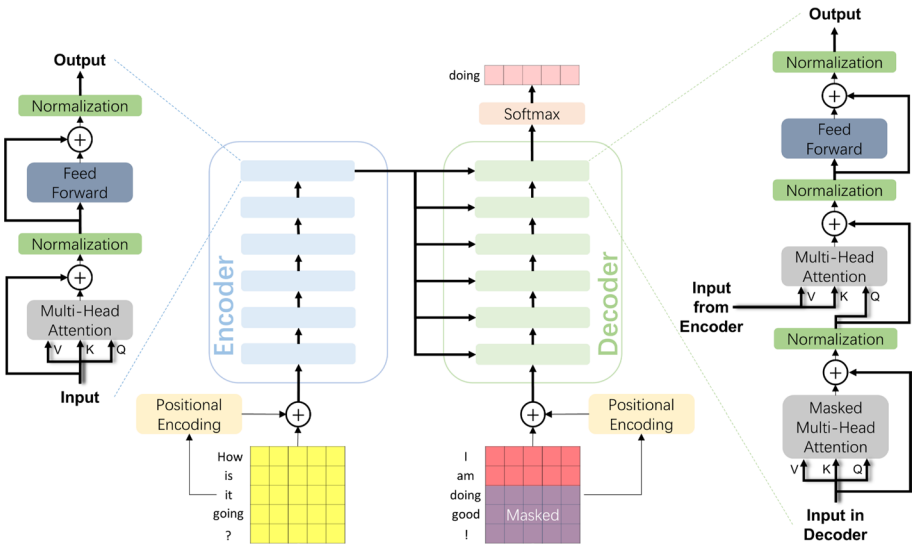

Fig. 15 Transformer computational mechanism

globally, Vaswani et al. (2017) intensively used the self-attention mechanism, a weighted average method to extract key features and fuse the focal information, to design the Transformer model of the encoder-decoder structure.

The general process of the transformer architecture is shown in Fig. 15. With receiving the vector representation of a whole sentence, the transformer model encodes the positional information into it to form the model input. The transformer encoder consists of six identical blocks whose operations are shown on the far left of Fig. 15. Each block includes a multi-head self-attention layer and a position-wise fully connected feed-forward layer. The multi-head self-attention comprises multiple self-attention sub-layers that share the same input. They can relate different positions of a single sequence using the transformation and operation of three matrices, $\boldsymbol{V}$, $\boldsymbol{K}$, and $\boldsymbol{Q}$. For each self-attention, $\boldsymbol{V}$, $\boldsymbol{K}$, and $\boldsymbol{Q}$ are derived from three different linear transformations of the same source, designed to capture the relationships between the input information.

When it comes to the transformer decoder, its structure is very similar to the encoder's, also consisting of six identical blocks, as shown on the far right of Fig. 15. The most significant difference is that each block in the decoder has two multi-head self-attention layers. And they are both slightly differentiated from those in the encoder. The first layer's input matrix is partially masked to prevent positions from attending to subsequent positions. For example, Fig. 15 illustrates the information state at a certain moment during the transformer training process, where the transformer model generates the word "doing" based on the input utterance "How is it going?" and the generated contents "I am" only. The masking operation in the first multi-head self-attention layer ensures that the decoder block cannot utilize the contents that will be produced. For the second layer, its matrices $\boldsymbol{V}$ and $\boldsymbol{K}$ are computed from the encoder output, while the matrix $\boldsymbol{Q}$ is derived from a previous layer in the block. The decoder with six identical blocks is required to capture the information extracted by the encoder and the contents produced by itself before the current moment.

A burgeoning paradigm of pre-training and fine-tuning in recent years further intensifies the propagation of transformer architecture. Fine-tuning is a common transfer learning method that partially adapts a model pre-trained on a large scale of datasets for a new scenario with similar data characteristics. It is particularly beneficial when the data volume is limited to prevent overfitting or in an application where the data amount is enormous to accelerate the model training process. In most cases, the success of a text-related task largely depends on the semantic feature mining from the given corpus. A model pre-trained on a large number of corpora can facilitate efficient text feature extraction by fine-tuning operations for transfer to new applications. Two well-acknowledged pre-trained transformer techniques are the Bi-directional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT). The more commonly used former proposed by Devlin et al. (2019) from the Google AI team is a bi-directional language model based on the transformer encoder structure, while the other provided by OpenAI (Radford et al. 2018) is essentially a transformer decoder that adopts the masked self-attention and can only embed one-side information for a word in the sentence. They are trained on massive unlabeled datasets and can be fine-tuned for various applications without extensive modifications to the specific architecture required for a given task.

Table 3 Summary of deep learning technologies and their usages

Usage technique	Natural language pre-processing	NLU		NLG			External knowledge enhancement
		Intent recognition and slot filling	Topic and question identification	Response scoring model	Response selection model	Response generation	
Standard DNN	2019: Aleedy et al. (2019), Xue et al. (2019), Prasomphan (2019a) and Hardalov et al. (2019) 2020: Jiao (2020) and Kushwaha and Kar (2020) 2021: Ferrod et al. (2021) and Brahma et al. (2021)	–	2018: Oh et al. (2018) 2019: Paul et al. (2019)	2019: Hardalov et al. (2019) and Kulkarni et al. (2019)	2021: Canas et al. (2021)	–	–

Table 3 (continued)

Usage technique	Natural language pre-processing	NLU		NLG			External knowledge enhancement
		Intent recognition and slot filling	Topic and question identification	Response scoring model	Response selection model	Response generation	
RNN	2017: Bartl and Spanakis (2017)	2018: Liao et al. (2018)	2019: Zhao et al. (2019)	2017: Li et al. (2017) and Bartl and Spanakis (2017)	2018: Singh et al. (2018)	2017: Xu et al. (2017) and Pradana et al. (2017)	2018: Specialized knowledge detection (Liao et al. 2018)
	2018: Moirangthem et al. (2018) and Quan et al. (2018)	2019: Xue et al. (2019) and Zhao et al. (2019)		2018: Yang et al. (2018)	2020: Franco et al. (2020)	2018: Ma et al. (2018), Aalipour et al. (2018) and Liao et al. (2018)	2019: Specialized knowledge detection (Olabiyi et al. 2019)
	2019: Kulkarni et al. (2019) and Prasomphan (2019b)	2020: Yu et al. (2020); Haihong et al. (2020) and Bhathiya and Thayasivam (2020)		2019: Prasomphan (2019a), Zhao et al. (2019) and Prasomphan (2019b)	2021: Li et al. (2021)	2019: Prajwal et al. (2019), Kang and Lee (2019), Sheikh et al. (2019), Olabiyi et al. (2019), Aleedy et al. (2019) and Chen et al. (2019)	2021: Emotion recognition (Chang and Hsing 2021; Majid and Santoso 2021; Tiwari et al. 2021; Zhang et al. 2021); Service mode classification (Ferrod et al. 2021; Lin et al. 2021b; Yang et al. 2021b)
	2020: Kushwaha and Kar (2020) and Ren et al. (2020)	2021: Majid and Santoso (2021) and Wu et al. (2021)		2020: Damani et al. (2020)		2020: Kushwaha and Kar (2020), Haihong et al. (2020), Nuruzzaman and Hussain (2020) and Ren et al. (2020)	
						2021: Kushwaha and Kar (2021), Lin et al. (2021b) and Chang and Hsing (2021)	

Table 3 (continued)

Usage technique	Natural language pre-processing	NLU		NLG			External knowledge enhancement
		Intent recognition and slot filling	Topic and question identification	Response scoring model	Response selection model	Response generation	
CNN	2018: Moirangthem et al. (2018) and Quan et al. (2018) 2019: Chen et al. (2019), He et al. (2019) and Prasomphan (2019b) 2020: Liu et al. (2020) and Ren et al. (2020) 2021: He and Tang (2021)	2017: Li et al. (2017)	2019: Kulkarni et al. (2019)	2018: Qiu et al. (2018) 2019: Prasomphan (2019a), Zhao et al. (2019) and Prasomphan (2019b) 2020: Song et al. (2020)	2019: Paul et al. (2019) 2021: Li et al. (2021)	2019: Aleedy et al. (2019) and Chen et al. (2019)	2021: Emotion recognition (Chang and Hsing 2021); Service mode classification (Lin et al. 2021b)
CapsNet	–	2021: Tiwari et al. (2021)	–	–	–	–	–
GNN	–	–	–	–	–	–	2021: Knowledge graph (Lin et al. 2021a)
GAN	–	–	–	–	–	2019: Olabiyi et al. (2019) 2020: Ren et al. (2020)	–

Table 3 (continued)

Usage technique	Natural language pre-processing	NLU		NLG			External knowledge enhancement
		Intent recognition and slot filling	Topic and question identification	Response scoring model	Response selection model	Response generation	
DRL	–	–	–	–	2017: Williams et al. (2017) 2019: Hatua et al. (2019) 2020: Zhao et al. (2020) 2021: Zhang et al. (2021)	2017: Kandasamy et al. (2017) 2018: Liao et al. (2018) 2020: Ren et al. (2020)	–
Transformer	2020: Damani et al. (2020) and Shalyminov et al. (2020) 2021: Yang et al. (2021b), Li et al. (2021), Yang et al. (2021a) and Chang and Hsing (2021)	2021: Tiwari et al. (2021), Yu et al. (2021) and Lothritz et al. (2021)	–	2020: Tahami et al. (2020)	2021: Li et al. (2021)	2020: Shalyminov et al. (2020)	2021: Service mode classification (Lin et al. 2021b)

Table 4 Summary of deep learning technological highlights and limitations

Technique	Feature							Highlight	Limitation
	NLP	NLU		NLG			EKE		
		①	②	③	④	⑤			
Standard DNN	✓		✓	✓	✓			Good at processing static data; Capable of handling large-scale network models and integrating all information to fit various data types and tasks	Poor performance in processing sequential and spatial data; Prone to overfitting; Difficulty in extracting more abstract features
RNN	✓	✓	✓	✓	✓	✓	✓	Good at processing sequential data and extracting temporal features	High computational complexity due to considering dependencies between time steps; Low training efficiency; Prone to gradient vanishing and explosion problems
CNN	✓	✓	✓	✓	✓	✓	✓	Good at extracting spatial features; Strong parallelism; High training efficiency	Certain limitations on the length and width of input data; Loss of position information in sequential data processing; Difficulty in handling long-range dependencies
CapsNet		✓						Good at handling hierarchical and spatial relationships with the concept of capsules that represent a feature with a vector	High computational complexity due to complex capsule structure and dynamic routing algorithm; Limited attempts in practice to examine the performance
GNN							✓	Good at processing graph data; Capable of capturing structural information between nodes and generating informative features	High computational complexity for large graphs
GAN						✓		Aimed at generating realistic samples in semi-supervised learning	Hard to train stably; Difficulty in handling discrete textual data
DRL					✓	✓		Good at sequential decision problems considering long-term income in a trial-and-error mechanism	High computational complexity due to the requirements of a lot of data, time, and computing resources; Low training and sampling efficiency due to frequent trials and feedback
Transformer	✓	✓		✓	✓	✓	✓	Good at extracting feature representations with contextual relevance and adapting to input sequences of different lengths; Strong parallelism; Global receptive field	Limited local information acquisition capability due to the unique position encoding method

NLP Natural Language Pre-processing (Word Embedding), *NLU* ① Intent Recognition and Slot Filling; ② Topic and Question Identification, *NLG* ③ Response Scoring Model; ④ Response Selection Model; ⑤ Response Generation, *EKE* External Knowledge Enhancement.

Table 5 Summary of paper contributions

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Bartl and Spanakis (2017)	RNN			RNN				A retrieval-based dialogue system utilizing utterance and context embeddings for customer services
Kandasamy et al. (2017)						DRL		A study of chatbots with RNN and DRL architectures when the rewards are noisy and expensive to obtain in the context of restaurant recommendations
Li et al. (2017)		CNN		RNN				A chatbot designed for creating an innovative online shopping experience in e-commerce
Pradana et al. (2017)						RNN		A chatbot developed to improve the interactivity and effectiveness of corporate website

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Williams et al. (2017)					DRL			A hybrid code network optimized with supervised learning or reinforcement learning to reduce the amount of training data in a customer-facing dialog system
Xu et al. (2017)						RNN		A chatbot for customer services showing empathy to help users on social media cope with emotional situations
Aalipour et al. (2018)						RNN		A Bi-RNN architecture customized to fit the domain-specific nature of enterprise customer support
Liao et al. (2018)		RNN				RNN, DRL	RNN	A multimodal fashion chatbot optimized with DRL to capture fine-grained semantics and generate responses

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Ma et al. (2018)						RNN		Tensor Encoder Generative Model collaborating data of many shops in customer service dialogue systems to alleviate the disadvantage of data insufficiency
Moirangthem et al. (2018)	RNN, CNN							A classification model to discriminate user utterances between task-oriented and chit-chat conversations
Oh et al. (2018)			Standard DNN					An out-of-domain detection method based on sentence distance for banking dialogue systems
Qiu et al. (2018)				CNN				Ae multi-turn conversation model based on CNN for context-aware question matching in e-commerce

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Quan et al. (2018)	RNN, CNN							A real estate chatbot with daily updated data of real estate information in Hanoi and Ho Chi Minh cities
Singh et al. (2018)					RNN			A chatbot using TensorFlow for small industries or business
Yang et al. (2018)				RNN				A learning framework that leverages external knowledge for response ranking in the context of technical support
Aleedy et al. (2019)	Standard DNN					RNN, CNN		A chatbot predicting a suitable and automatic response to customers' queries
Chen et al. (2019)	CNN					RNN, CNN		A review-driven framework of answer generation for product-related questions in e-commerce

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Hardalov et al. (2019)	Standard DNN			Standard DNN				A deep neural architecture from the domain of machine reading comprehension to re-rank the suggested answers from different models using the question as a context
Hatua et al. (2019)					DRL			A goal-oriented chatbot using transfer learning and attention mechanism for movie ticket booking
He et al. (2019)	CNN							A model based on recurrent pointer networks aligning question and answer utterances in customer services
Kang and Lee (2019)						RNN		A context-aware dialog generation system through external memory for chit-chat conversations

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Kulkarni et al. (2019)	RNN		CNN	Standard DNN				A question–answer matching framework to answer both factoid and non-factoid user questions on product pages
Olabiyi et al. (2019)						RNN, GAN	RNN	A persona-based multi-turn conversation model in an adversarial learning framework
Paul et al. (2019)			Standard DNN		CNN			A focused domain contextual chatbot framework for customer services in resource-poor languages
Prajwal et al. (2019)						RNN		A universal semantic web assistant based on a sequence-to-sequence model
Prasomphan (2019a)	Standard DNN			RNN, CNN				A retrieval-based method for chatbot improvement in trading systems for SMEs

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Prasomphan (2019b)	RNN, CNN			RNN, CNN				A prototype combining retrieval and generative methods in trading systems for SMEs
Sheikh et al. (2019)						RNN		A generative model for Human Resource
Xue et al. (2019)	Standard DNN	RNN						An agent-assist chatbot boosting the effectiveness of customer support agents
Zhao et al. (2019)		RNN	RNN	RNN, CNN				A chatbot providing instructional answers from a knowledge base in mobile customer services
Bhathiya and Thayasivam (2020)		RNN						A meta-learning method for few-shot joint intent detection and slot-filling
Damani et al. (2020)	Transformer			RNN				Optimized Transformer models for FAQ answering

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Haihong et al. (2020)		RNN				RNN		A multi-domain chatbot delivering or requesting information according to specific user requests
Franco et al. (2020)					RNN			A business-driven chatbot for cybersecurity planning and management
Jiao (2020)	Standard DNN							A financial chatbot based on entity extraction using RASA NLU and neural network
Kushwaha and Kar (2020)	RNN					RNN		A language model-driven chatbot for businesses to address marketing and selection of Products

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Liu et al. (2020)	CNN							Gated attentive CNN dialogue state tracker utilizing the gated attentive convolutional encoder and introducing historical information
Nuruzzaman and Hussain (2020)						RNN		A chatbot for the insurance industry that uses multiple strategies to generate a response
Ren et al. (2020)	RNN, CNN					RNN, GAN, DRL		An end-to-end system to tackle the task of conversational recommendation with an adversarial reinforcement learning approach to refine the quality of generated system actions adaptively
Shalyminov et al. (2020)	Transformer					Transformer		A hybrid generative-retrieval model able to perform both response generation and ranking

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Song et al. (2020)				CNN				A triple CNN model for retrieval-based question answering system in e-commerce
Tahami et al. (2020)				Transformer				A cross-encoder architecture that transfers knowledge from one model to a bi-encoder model using distillation
Yu et al. (2020)		RNN, CNN						A joint model based on intent information enhancement for multi-domain language understanding
Zhao et al. (2020)					DRL			A Dynamic Reward-based Dueling Deep Dyna-Q model that can learn policies in noise robustly

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Brahma et al. (2021)	Standard DNN							A named entity recognition approach to identify the food quality descriptors from a given message
Canas et al. (2021)					Standard DNN			A data-driven dialog management technique providing flexibility to develop, deploy, and maintain the dialog module in commercial platforms
Chang and Hsing (2021)	Transformer					RNN	RNN, CNN	An emotion-infused deep neural network for emotionally resonant conversation
Ferrod et al. (2021)	Standard DNN						RNN	A classification model to identify users' domain expertise from dialogues in Telco commerce

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
He and Tang (2021)	CNN							A method of context representation learning on sequential data for dialogue state tracking
Kushwaha and Kar (2021)						RNN		A language model-driven chatbot for interactive marketing in the post-modern world
Li et al. (2021)	Transformer				RNN, CNN, Transformer			A deep context modeling method for multi-turn response selection in dialogue systems
Lin et al. (2021a)							GNN	A personalized entity resolution method with dynamic heterogeneous knowledge graph representations
Lin et al. (2021b)						RNN	RNN, CNN, Transformer	A predictive approach for Wait-or-answer tasks in e-commerce dialogue systems

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Lothritz et al. (2021)		Transformer						A comparative study exploring multilingual and multiple monolingual models for intent classification and slot filling in banking
Majid and Santoso (2021)		RNN					RNN	A conversation sentiment and intent categorization method using context RNN for emotion recognition
Tiwari et al. (2021)		CapsNet, Transformer					RNN	A dynamic goal-adapted task-oriented dialogue agent in mobile selling-buying scenarios
Wu et al. (2021)		RNN						A joint model of intent classification and slot filling for online customer services

Table 5 (continued)

Paper	Contribution							
	NLP	NLU		NLG			EKE	Artifact design
		①	②	③	④	⑤		
Yang et al. (2021a)	Transformer							An intelligent cloud customer service system based on tag recommendation
Yang et al. (2021b)	Transformer						RNN	A model predicting users' abandonment of a task-oriented chatbot service using explainable deep learning
Yu et al. (2021)		Transformer						A financial service chatbot based on deep bidirectional Transformers
Zhang et al. (2021)					DRL		RNN	An Emotion-Sensitive Deep Dyna-Q model for task-completion dialogue policy learning

NLP Natural Language Pre-processing (Word Embedding), *NLU* ① Intent Recognition and Slot Filling; ② Topic and Question Identification, *NLG* ③ Response Scoring Model; ④ Response Selection Model; ⑤ Response Generation, *EKE* External Knowledge Enhancement.

4 Summary of deep learning applications in business chatbots

With the understanding of various deep learning structures and their computational methods, we further analyze how each stream of deep learning technology can be used in dialogue systems and compare their characteristics, as summarized in Tables 3 and 4. Table 5 presents selected papers' adopted deep learning streams in each application and concludes their artifact design. This section provides a thorough summary and critical discussion on the deep learning applied in business chatbots from four critical perspectives: pre-processing for natural languages, NLU, NLG, and external knowledge enhancement for response quality improvement.

4.1 Pre-processing of natural languages

Pre-processing is a series of adjustment operations to convert natural language text into an analyzable and predictable computer language form, including spelling correction, tokenization, stop word removal, word normalization, and text vectorization (word embedding). Traditional pre-processing technologies are based on the intuitive understanding of human languages and statistical learning methods (Johnson 2009). For computer processing, designers would program to split the words, remove the unimportant high-frequency terms, convert words to lowercase, change parts of speech, and conduct many other operations. The texts are eventually mapped to vector representations through bag-of-words or n-gram statistical language models before further NLU or NLG processing. This process of vector representation of words is called word embedding. However, not all tasks require the same level of pre-processing operations, and previous methods might ignore the order of words, part-of-speech, and synonyms, resulting in a massive loss of information. The current increase in deep learning technologies provides more options for the word embedding process of NLP. Novel text vector representation techniques in neural network structure reduce the tedious step-by-step pre-processing, replace the traditional statistical learning methods, and overcome the strict demand of statistical learning for elaborate feature engineering. The performance of chatbots has benefitted from the leap in deep learning. It has improved significantly from the pre-processing stage by capturing more information from human language, which further promotes chatbot upgrading. The following contents elaborate on how developers choose deep learning to pre-process received input texts for dialogue systems in consideration of development requirements and technical features.

One typical neural network embedding technology is Word2vec, proposed by Mikolov et al. (2013). It is a group of techniques that builds a standard neural network structure model to reconstruct linguistic word texts and detect synonymous words or suggest additional words for a partial sentence. The by-products of the Word2vec model are used to generate word vectors. The essence of this technology is to reduce dimensions for the one-hot vector of a word through the parameters of the input or output layer of a shallow, double-layer neural network, which significantly improves the performance of computing continuous vector representations of words from huge datasets. Many business chatbots adopted this technology as the standard pre-processing of text vectorization for the received user input, such as customer service support chatbots on Tweet (Aleedy et al. 2019) or in call centers (Xue et al. 2019), a train ticket trading system (Prasomphan 2019a), and an online stock information provider (Jiao 2020). Other technologies, such as GloVe developed by Pennington et al. (2014) at Stanford and fastText offered by Facebook's AI

Research lab (Bojanowski et al. 2017), apply a similar standard neural network structure while exploring more relationships among words. GloVe considers the global word-word co-occurrence statistics of the training corpus for word embedding, in which the distance between words implies more semantic information. This technology has been utilized in common chatbots devoted to customer support services on social media for marketing needs (Hardalov et al. 2019; Kushwaha and Kar 2020). FastText aggregates the n-gram model for word order information representation and the subword model (a text processing method between word and character levels) for atomic unit segmentation of word vectors. Its feasibility has been tested under e-commerce scenarios, such as the commercial telco platform (Ferrod et al. 2021) and the online food delivery company (Brahma et al. 2021).

The convolutional and pooling operations allow CNN to perform well in dealing with two-dimensional data. At the same time, some researchers have attempted to adapt the structure for the one-dimensional convolutional model to extract text features. For instance, Chen et al. (2019) combined CNN with attention and gate mechanisms to extract product review features and encode product-related questions in an e-commerce application. Liu et al. (2020) used a similar technique to learn the dialogue sequence representation. In other customer service scenarios, such as hotel and restaurant consulting, developers (He et al. 2019; He and Tang 2021) have also attempted to use CNN to encode conversation information. In particular, CNN can produce embedded information at the character level. This advantage of CNN is conducive to solving the out-of-vocabulary problem. One well-accepted technique is Char-CNN proposed by Zhang et al. (2015), and Quan et al. (2018) used it to process dialogue information of a chatbot built for the real estate industry.

RNN and its variants, such as LSTM and GRU, are created to handle sequence data, which is naturally suitable for dialogue information processing as human languages are sequential. The internal state (memory) structure can help the system learn the nonlinear features of input sentences, integrate the order of words, and process variable-length sequences, which solves the polysemy problem, earning the interest of many dialogue system developers. In Word2vec, the order of words is not emphasized under the assumption of the bag-of-words model, but it is crucial in the temporal dynamic recurrent network. Because of the powerful text parsing performance of the Seq2seq (encoder-decoder) structure, encoders with different processing units are widely used for sentence embedding. Bartl and Spanakis (2017) and Kushwaha and Kar (2020) used the RNN encoder to extract context features and embed dialogues to build customer service supporting social media chatbots. Primitive encoders only encoded the sentence in a fixed order, sequentially or reversely, to process the text word by word. The improved RNN techniques with the bi-directional structure have proved more effective (Zhou et al. 2016) and have been extensively used in NLP. ELMo (Peters et al. 2018) is one of the representative bi-directional language models to create contextualized word embedding. This architecture can capture more text features by reading the input sentence from two directions, which means the content before and after a word can be processed simultaneously. Variants such as Bi-LSTM and Bi-GRU are common for the dialogue embedding in auto-response applications, including the service industry (Quan et al. 2018; Ren et al. 2020) and e-commerce platform (Kulkarni et al. 2019).

The essence of neural network embedding technologies is to leverage the network structure characteristics to explore the relationships between words in a given text. Each network can extract distinct text features, although not necessarily effective. A hybrid method might be feasible for complex text processing to capture multiple text features. For example, CNN is good at dealing with image data but did not improve much when representing one-dimensional text information. However, when combined with RNN, it can help

integrate various extracted text features. Moirangthem et al. (2018) introduced a hybrid of CNN and GRU and evaluated its performance in modeling longer semantic sequences. Prasomphan (2019b) combined CNN with RNN to extract the underlying abstract features of data for dialogue representation in an online sales assistant application system. Ren et al. (2020) adopted a similar method to extract n-gram features from conversations and learn the synergic representation in a restaurant conversational recommender system.

Since the proposal of transformer architecture by the team from Google Brain (Vaswani et al. 2017), its astonishing performance in handling sequential data has induced a boom in replacing RNN-based models to achieve various NLP tasks, even extending to other research fields, such as computer vision. The full use of the encoder-decoder structure and self-attention mechanism makes it more efficient for transformers to process a sentence parallelly and globally than RNN models with a word-by-word mode. Transformer embedding can provide each text input word vector with the global position and context information. BERT and GPT are the most frequently mentioned transformer embedding techniques in up-to-date business chatbots. They have attracted considerable attention in the latest chatbot design and have been rapidly utilized in various business scenarios, including multi-domain FAQs for small and medium enterprises (SMEs) (Damani et al. 2020; Shalyminov et al. 2020), customer services in the banking industry (Yang et al. 2021b) or e-commerce platforms (Yang et al. 2021a; Li et al. 2021), and emotionally resonant conversation (Chang and Hsing 2021). Observing an increasing number of studies using transformer technology, we believe it will continue to dominate the NLP field for a while.

4.2 Natural language understanding

NLU refers to the challenging task of endowing a machine (computer) with the reading comprehension capability of human languages, including making a computer understand natural languages and confirming its realization. It is a common, necessary component in pipeline chatbot design to maintain human–computer conversations and recognize the primary semantic information based on pre-processed user input. Previous practices were closely related to pattern-matching techniques. An NLU module must acknowledge the possible sentence pattern in the user utterance and determine the most matched predefined one for the subsequent NLG. The matching process was usually string-level and widely used the distance comparison between two sentences (e.g., cosine similarity) as the matching similarity (Thomas 2016; Pilato et al. 2007; Wei et al. 2014; Setiaji and Wibowo 2016; Augello et al. 2009). Currently, the modularization design of the pipeline structure demands refining the extracted knowledge and capturing more information from natural languages. Deep learning technologies can produce extraordinary performance in such fine-grained mining of sentence semantic information. Unlike matching a similar sentence structure to give a preset response, the NLU module in the current pipeline structure will perform several subtasks to generate more granular semantic content, leading to more accurate answers.

4.2.1 Intent recognition and slot filling

The most commonly detected information includes user intents and slots concerned with the predefined entities. The intents and slots are template information designed in advance. Intent recognition refers to classifying the user intent in the utterance. Slot filling detects the possible entity values and matches the correspondent entity type from the utterance.

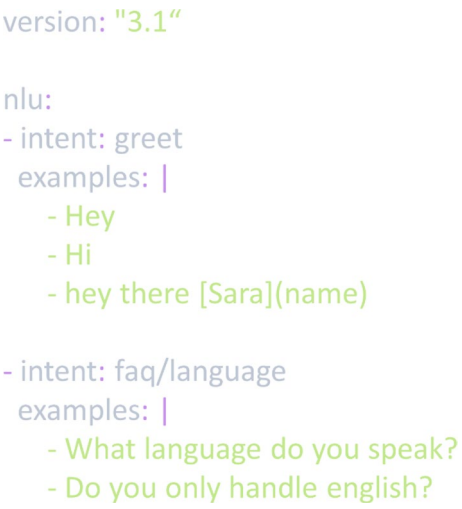

Fig. 16 Short example of intents and slots defined in RASA 3.1. *Source* rasa.com/docs/rasa/training-data-format/#example

Figure 16 illustrates the intents and slots in the model training corpus of the RASA Version 3.0 NLU component. The intent "greet" includes three kinds of greeting ways in which the third example involves the entity "Sara" that belongs to the entity type "name." These corpora are designed to train the NLU model. In practical application, the trained NLU component will classify a user greeting as "greet" intent and try to detect the entity (username) to fill in the "name" entity type.

Existing literature deals with these two classification subtasks in two modes, separately or jointly. For orderly detection, the intent recognition will usually be conducted independently before the slot filling. Similar to the technique preference in word embedding, CNN- and RNN-based deep learning models were widely used to figure out the relationships and features among the user input and the predefined intents and slots because of their outstanding data processing ability. Li et al. (2017) used a one-layer convolution-pooling CNN to classify user intents in an e-commerce service assistant. LSTM or its well-known variant, Bi-LSTM, have been frequently applied to recognize the intents in customer service scenarios, such as fashion need analysis (Liao et al. 2018), call centers (Xue et al. 2019; Zhao et al. 2019), and other commercial applications (Majid and Santoso 2021; Arsovski et al. 2019). Regarding slot filling, researchers regard it as a sequence labeling task naturally suitable to adopt an improved RNN model. For example, the Bi-LSTM model could be used to achieve the task based on intent classification (Yu et al. 2020) or independent of intent classification (Haihong et al. 2020).

Some researchers prefer a joint training method for its advantages of utilizing the dependency between intents and slots (Liu and Lane 2016; Xu and Sarikaya 2013; Hakkani-Tur et al. 2016), which appears to be closer to how the information flows in human brains with semantic hierarchy exploited (Zhang et al. 2018a). For example, the sentences "*Welcome to New York* is a marvelous song" and "Welcome to New York, babe" have the same strings, "Welcome to New York", but their intents are intuitively different based on our understanding. Slot types may be helpful for a computer to understand the intents of human languages. The slot in the first one could be recognized as a "song" slot with the value of "Welcome to New York," while a "location" slot could be

extracted from the second one with the value of "New York". Their intents should be classified separately according to different slot values because of the discussion about a song and the welcome greeting. Deep learning technologies based on RNN, especially the improved LSTM and GRU, were once more the common choices for chatbot designers (Bhathiya and Thayasivam 2020; Wu et al. 2021) to complete the joint task of intent and slot detection. In the last two years, the potential of transformer architecture has been exploited greatly in NLP, and the mode of pre-training and fine-tuning makes BERT applicable for various tasks related to text features, including this joint detection task. It can appear in scenarios such as mobile selling-buying in an integrated form of BERT and CapsNet (Tiwari et al. 2021), financial investment (Yu et al. 2021), and banking (Lothritz et al. 2021) for customer services.

4.2.2 Topic domain and question type classification

To improve the responding performance of pipeline structure, some researchers have tried coming up with more refined subtasks to help the chatbot obtain a better understanding of natural languages. Apart from intent and slot detection, chatbot developers seek to produce more semantic information through the topic classification of a user utterance to increase the accuracy of downstream tasks. Oh et al. (2018) used a fully connected neural network to classify the dialogue domain to detect out-of-domain utterances in banking services. Similarly, Paul et al. (2019) applied a neural network to classify topics in given texts for the services in an electric shop. A pre-division of user questions might be another feasible way of improving reading comprehension. Kulkarni et al. (2019) developed a CNN model to classify e-commerce question types for customer services. Zhao et al. (2019) adopted a hybrid model of CNN and LSTM to implement a question category classifier in mobile customer services.

4.3 Natural language generation

NLG is the fundamental function of a chatbot that produces responses that interact with users. It is the last component in the dialogue systems of pipeline structure and the only essential constitution in end-to-end design. Response-producing methods can be categorized into two types from an acknowledged technical point of view: retrieval- and generation-based. When receiving user input, responses given in the retrieval-based methods are predefined, while those provided with the latter methods are generated word- for-word. Furthermore, based on our observations from the literature of the last five years, retrieval-based techniques can be refined further as scoring models and response action selection regarding the utilization of the preset responses. This difference in model input and output design affects developers' preference for deep learning technologies in the response retrieval function. Accordingly, we summarize these three response-producing ways of NLG to conduct comprehensive comparative analysis in this section: (1) scoring model to rank response candidates, (2) classification model for response selection, and (3) response generation.

4.3.1 Response scoring model

A scoring model is designed to calculate a matching score for evaluating a preset response filtered by the NLU component (if any). The model inputs generally include

the pre-processed representations of the user utterance, a possible response, and the chat context or dialogue history. Then, the model computes a score to measure the matching degree between given contexts and the response for all possible candidates. The chatbot will select the one(s) with the highest score from the response candidates and react to continue the conversation with users. Two formats of how a predefined response can be represented as input to the model are using the "answer" of given contexts or the "question" in "question–answer" pairs. A dialogue system in the previous format replies to users with the input "answer" directly, while that in the latter form will use the "answer" in "question–answer" pairs based on the correspondent "question." These two ways share a similar design philosophy and technique principle. An effective scoring model is expected to learn the features of the input contexts and responses and capture the relationships among them. The score can be the similarity (Bartl and Spanakis 2017; Prasomphan 2019b; Zhao et al. 2019), probability (Shukla et al. 2020; Hardalov et al. 2019), matching degree (Yang et al. 2018; Qiu et al. 2018; Prasomphan 2019a), confidence score (Li et al. 2017), and other metrics to evaluate the extent to which the response fits the correspondent user utterance.

The CNN- and RNN-based deep learning technologies or their hybrids are not unexpected to be frequently used for complex text feature learning. Seq2seq and bi-directional structures of RNN-based improvements remain the most common choices for generating the evaluation score in customer service scenarios such as e-commerce (Li et al. 2017), Microsoft technical support (Yang et al. 2018) and other applications (Bartl and Spanakis 2017; Damani et al. 2020). CNN-based models integrated with strategies such as attention mechanism and transfer learning also apply in e-commerce services (Qiu et al. 2018; Song et al. 2020). The performance of hybrid models has been demonstrated in different combinations, including CNN and RNN in SME trading systems (Prasomphan 2019b), CNN and LSTM in mobile customer services (Zhao et al. 2019), and CNN and GRU in a train trading system (Prasomphan 2019a).

RNN- and CNN-based deep learning technologies have been commonly used in the pre-processing and NLU stages of extracting semantic features for input texts, and hence, some researchers might try other types of neural network models in the NLG process to learn natural language information that differs from that captured by CNN- or RNN-based models. The standard neural networks with structural fine-tuning can also effectively calculate ranking scores. Hardalov et al. (2019) improved the neural network model to predict the probability of response candidates in the simulation of Apple's customer support on Twitter. Kulkarni et al. (2019) built a neural network for similarity-based response ranking in e-commerce services. With the advancements in transformer technology, its pre-training and fine-tuning schema has been utilized to upgrade a similar function. For example, Tahami et al. (2020) developed a scoring model in BERT to accelerate the computation of the matching degree between any new conversation history and a response candidate.

4.3.2 Response selection model

Compared to the scoring model using responses as input, a classification model for response selection takes responses or response-producing ways as output with user utterance and chat contexts as input. A dialogue system in such classification methods reacts to users by selecting the most probable output action (a response or a way to produce the response) according to the classification results (usually a probability distribution over actions). This approach is common in small and medium-scale application scenarios where the conversation domain and response type are finite.

Almost all typical neural network models are capable of the response classification task. In the context of customer services for small businesses, Singh et al. (2018) and Paul et al. (2019) adopted RNN and CNN, respectively, to classify the "tags" that can uniquely identify the correspondent "pattern-response" pairs. With the support of the DialogFlow framework, Canas et al. (2021) added a standard neural network for response selection in e-commerce services, and Franco et al. (2020) created an LSTM model for a similar purpose with another popular framework, RASA, in the scenario of cybersecurity-related queries. They all treated each chatbot response as action and used a neural network model to select the most suitable one based on the NLU information processed by the pipeline design. The fashionable transformer technique with the pre-training mechanism was applied to this task. Li et al. (2021) combined RNN, CNN, and BERT to improve the response selection performance of multi-turn conversations in e-commerce.

Because of the technical characteristics of the response classification task, the quantity of the responses (actions) is fixed, and each unit in the output layer of neural network models uniquely corresponds to a predetermined action, enabling it to leverage the value-based DRL at an advantage in long-term (multi-turn) performance. DQN is a reinforcement learning algorithm integrated with a deep learning model that calculates a value for measuring the long-term reward of the corresponding response. It has been applied in the scenarios of restaurant services (Williams et al. 2017), movie booking (Hatua et al. 2019), and Microsoft's customer support (Shukla et al. 2020). Researchers have also improved the DQN-based algorithm to fit specific application scenarios, such as the Dynamic Reward-based Dueling Deep Dyna-Q to mitigate the negative impact of data noise (Zhao et al. 2020) and the Emotion-Sensitive Deep Dyna-Q to provide emotion-related immediate feedback (Zhang et al. 2021) in the movie-ticket booking communication. They significantly improve users' interactive experience in long-term transactions.

4.3.3 Response generation

Compared to the previous retrieval-based response-producing methods, generation-based methods are those technologies that can "generate" responses without a searching step when receiving user utterances. Generation-based technologies imply that the neural network model can generate a brand-new response word-for-word. Theoretically, a dialogue system built with generation-based techniques has the potential to answer any query compared to retrieval-based dialogue systems that rely on predefined corpora to give responses.

The rise of these generative methods in chatbots follows the development of RNN-based models proficiently processing sequence data. Among the RNN variants, LSTM has been the most preferred neuronal structure in recent years. With the enhancement of the Seq2seq design, the generative operation mode of RNN has gradually evolved into a universal architecture, as shown in Fig. 8. A Seq2seq model built with LSTM for generating responses can be extensively found in the customer services of social media (Xu et al. 2017; Kushwaha and Kar 2020, 2021), web assistants (Pradana et al. 2017; Prajwal et al. 2019), chit-chat (Kang and Lee 2019), insurance consultancy (Nuruzzaman and Hussain 2020), e-commerce (Lin et al. 2021b), and many other scenarios (Ma et al. 2018). LSTM can deal with long-range dependencies of the input sequence to avoid the tendency of stressing recent contextual information. Improvements with bi-directional design, such as Bi-LSTM and Bi-GRU, can be adopted to increase the extracted sequence information from two directions (Aalipour et al. 2018; Sheikh et al. 2019; Haihong et al. 2020; Chang and Hsing 2021). Seq2seq allows variable-length sentences as input and output to simplify

the processing of model training data. Its upgrade, HRED, can consider the turn-taking nature and perform better in multi-turn conversations (Olabiyi et al. 2019; Liao et al. 2018; Bartl and Spanakis 2017). The feasibility of all these improvements has been examined in applications from large enterprises to small shops, and the Seq2seq-based design has also become a classic generation-based artifact.

RNN and its variants can be further improved by integrating other neural networks and advanced technologies to build a satisfactory generation-based dialogue system. Similar to other applications of deep learning in text processing, CNN is the most common combination with RNN or Seq2seq-based models. For example, Aleedy et al. (2019) integrated CNN, LSTM, and GRU into a hierarchical dialogue system design for customer support services on Twitter, while Chen et al. (2019) incorporated the Gated CNN into the Seq2seq model to build an e-commerce chatbot in cellphone and household electrics contexts. Utilizing these techniques enables learning the different text features with the support of various network structures. GAN is another common choice integrated with RNN-based models to generate from text to text with a similar structure. This technique can enhance the original data characteristics with text retelling and improve the robustness of unknown data by data augmentation. Olabiyi et al. (2019) used it to capture and shrink the syntactic and semantic difference between the ground truth and the generated persona-specific response. Similarly, Ren et al. (2020) designed a GAN in the dialogue system of restaurant services to make the response generated by Bi-LSTM close to that of high quality produced by humans.

The progressive technologies of transformer and DRL have also been examined in the generative task. The transformer's parallel computing capability in global sequence processing greatly speeds up the computation of encoder-decoder architecture compared to RNN models, and its combination with pre-training and fine-tuning mechanism can help with the data efficiency problem. For example, Shalyminov et al. (2020) employed the transformer technique GPT-2 in a multi-domain chatbot design to fit the dialogue system into a rapid prototyping cycle for new products.

Another advanced technology, DRL, can be used to train the model to favor generated responses with high long-term rewards instead of being stuck in universal and repetitive words such as "I don't know." Generation-based methods can be regarded as models that can produce infinite response actions; hence, the responses need to be evaluated by continuous indicators with a limited range of values to design the rewards. The rewards are made to adjust the probability that the model generates a response. Choosing the policy gradient algorithm to help train such policy-based DRL design that can generate a high-quality response is appropriate to facilitate the continuation of the conversation. Kandasamy et al. (2017) and Liao et al. (2018) applied this technology to improve the user experience in recommendation services, and Ren et al. (2020) employed it in a generative ensemble model of Bi-LSTM and GAN for serving restaurant customers.

4.4 External knowledge enhancement

To enrich diversity and improve the relevance of responses, some researchers utilize external knowledge apart from the given contexts and utterances to assist in response production. A common practice based on deep learning is introducing extra memory space to build a neural network and training it to learn and store task-related knowledge. This additional knowledge needs to be provided artificially to train the neural networks. The models are required to derive more accurate and valuable content from user utterances and enhance

the performance of downstream identification and generation components with the additional learned knowledge.

One trending approach is infusing user emotion recognition into the NLU process to optimize the response performance regarding anthropomorphism. Researchers considered empathetic conversation an interactive service of high quality in business situations (Chang and Hsing 2021). They believed that user emotions could reflect the performance of dialogue systems and thus have made various attempts to incorporate human emotion detection into conversational agents and improve their anthropomorphism. Majid and Santoso (2021) added a sentiment classification function in the NLU process to provide satisfactory services. Chang and Hsing (2021) integrated a CNN into Bi-LSTM for emotionally resonant response generation to recognize user emotions, which were further infused into the generating process. Detected emotions can be utilized as indicators for user desire measurement to design the reward in DRL technology, which could speed up the agent learning human emotional feedback. For example, Tiwari et al. (2021) and Zhang et al. (2021) designed immediate rewards based on the numerical representation of contextual sentiments detected by GRU or LSTM models to achieve an instant evaluation of response actions in transaction services.

In addition to the universal sentiment knowledge, some attempts at the utilization of specialized external knowledge have been made for specific commercial applications., Some researchers tried to incorporate a topic domain-related recognizer into the NLU component to improve the user service experience from the relevance of responses. In transaction services, Liao et al. (2018) leveraged an HRED model to provide tips on product styles satisfying customers' fashion needs. Nangoy and Shabrina (2020) utilized a CNN model to classify the product type in user queries. For persona-specific conversations, Olabiyi et al. (2019) introduced and upgraded the HRED model integrated with GAN to learn the features of speaker personality, location, and sub-topic and disambiguate them before the process of generating responses.

Some additional knowledge can be brought in to increase the efficiency of automated conversation services. Ferrod et al. (2021) designed a bi-directional RNN-based model to match an appropriate expertise level with user demands in telco commerce. Lin et al. (2021b) added a text classifier to decide whether to wait for user input or give a response during the e-commerce conversation. Yang et al. (2021b) adopted an LSTM to predict user abandonment behaviors to switch to other modes of banking services timely.

The external knowledge can be represented in a high-dimensional form to facilitate pertinent conversations. Lin et al. (2021a) developed a GNN to jointly learn the customer and product embedding from a customer-product knowledge graph, which was expected to enhance the conversation connections between the customers and their target products in shopping guidance scenarios. The above attempts can effectively improve the performance of downstream tasks and optimize the self-service experience for users, which inspires us to enrich and intensify NLU functions by magnifying the multidimensional exploitation of available knowledge.

5 Critical analysis of the characteristics of business dialogue systems

Section 2 divides contemporary dialogue systems of chatbots appearing in academic business studies into two categories (pipeline and end-to-end chatbot) based on the building architecture. Combined with the acknowledged taxonomy from the NLG perspective, we

Table 6 Classification of contemporary dialogue systems and their characteristics

Characteristic	Category			
	Pipeline architecture		End-to-end architecture	
	Retrieval method	Generative method	Retrieval method	Generative method
Response source	Preset corpus (template, dataset, database)	Generative model trained with corpora	Preset corpus (template, dataset, database)	Generative model trained with corpora
Pre-processing	Statistical learning methods, Neural network embedding	Neural network embedding	Neural network embedding	Neural network embedding
NLU extracts	Intent, slot, topic domain, question type	Intent, slot	–	–
NLG	Scoring model, response classification model	Response generation	Scoring model, response classification model	Response generation
Context-sensitive enhancement	Context embedding, sentiment, personality, topic-related knowledge	Context embedding, emotion, topic-related knowledge	Context embedding	Context embedding
Extensibility of additional functionality	High: for framework-based, medium: for self-building	Medium	Low	Low
Data volume demand	Small, medium, and large scale	Medium and large scale	Small and medium scale	Medium and large scale
Dialogue scalability	Limited: added responses need to be compatible with other components	Scalable: the NLU component can be slightly adjusted to match added responses	Scalable: for scoring models, limited: for response classification models	Scalable: responses can be added to the training corpus
Responding stability	High	Low	Medium	Low
Difficulty of development	Low: for framework-based, Medium: for self-building	Medium	Low	Low
Corpus update and maintenance	Components need to be coordinated	The generative model needs to be trained	The whole model needs to be trained and adjusted	The whole model needs to be trained
Manual intervention level	High: for framework-based, Medium: for self-building	Medium	Low: for scoring models, Medium: for response classification models	Low
Expected practicability	High	Low	Medium	Low

Table 6 (continued)

Characteristic	Category			
	Pipeline architecture		End-to-end architecture	
	Retrieval method	Generative method	Retrieval method	Generative method
Advantage/potential	Each component can be independently adjusted and evaluated	New responses can be generated	Simple to build without a large dataset	New responses can be generated
Disadvantage/limitation	Error propagation, Manual intervention	Difficult to generate exact responses	Difficult to deal with multi-topic demands	Very difficult to generate exact responses
Application scenario	E-commerce, Social media, Banking, Technical support, etc.	E-commerce	E-commerce, Technical support	E-commerce, Social media, Technical support, Chitchat, etc.
Research gap	Unexplored application scenarios and commercial values Unclear user behaviors or psychological activities Unarticulated applicability and usability of diverse architectures A unified chatbot ecosystem			

further differentiate the chatbots regarding the retrieval-based and generation-based deep learning methods and conduct a comprehensive analysis of business chatbots in the four categories, as summarized in Table 6. Whether in the pipeline or end-to-end structure, a dialogue system employed with different deep learning methods in NLG has distinct technology choices and realization characteristics. We believe a critical comparison of technical features would provide detailed insights into the construction of business chatbots in the era of flourishing deep learning technologies. In this section, we illustrate the characteristics of each category and compare their existing application scenarios.

5.1 Chatbot in pipeline architecture

Chatbot's built-in pipeline architecture presents an explicit information processing flow for user utterances. Its NLU component extracts the knowledge from pre-processed natural languages, while the NLG component receives the NLU extracts and produces the response considering dialogue contexts and states. Each component can be adjusted and improved independently based on the designers' requirements, and the NLU component is a signature ingredient in this structure. Although the pipeline design requires more human manipulation and suffers the unavoidable error propagation issue among components, it is beneficial to distributed processing and validation of effectiveness for independent optimization of each component. In addition, the eventual pipeline design can be further differentiated due to the technical characteristics of different NLG technologies.

5.1.1 Pipeline architecture with retrieval methods

Most chatbots in pipeline design adopt a retrieval method (scoring model or response action selection) to determine a predefined response. Developers need to preset all the replies corresponding to the specified questions and segment conversational knowledge that each component is supposed to capture at different stages. Competent deep learning methods are selected to exhibit the extraction capability of their expert features and learn the exclusive knowledge in the assigned NLU or NLG tasks. This approach is particularly suited for application scenarios where collected dialogue data are limited, with the chatting domain and query type easily identified. The entire design is intuitive and easy to adjust and part, even in large-scale applications. As the volume of data increases, so does the manual operation and system maintenance workload.

Many online shopping platforms have created pipeline-based chatbots responding with retrieval methods. Chatbot developers design a deep learning classification model to recognize the intents, slots, query patterns, product types, emotional states, or other prerequisite knowledge and train another model to determine the most suitable response candidate in a regression or classification manner (Li et al. 2017; Paul et al. 2019; Kulkarni et al. 2019; Nangoy and Shabrina 2020; Tiwari et al. 2021; Lin et al. 2021a; Canas et al. 2021). In particular, Singh et al. (2018) applied the machine learning software library TensorFlow to demonstrate a neural network instance that ranked the response candidates filtered from the preset JSON file for feasibility in small business applications. Interestingly, some researchers (Hatua et al. 2019; Zhao et al. 2020; Zhang et al. 2021) have examined the utilization of DQN-based response selection methods and evaluated their availability to consider long-term effects in a movie-ticket booking scenario where multiple rounds of interaction and frequent information queries exist.

Another common commercial use is technical support services. Chatbots are expected to provide customers with instructional answers to technical problems that might otherwise require massive time, money, energy, or material costs. In mobile services, Xue et al. (2019) and Zhao et al. (2019) improved LSTM models adapted to match the most semantically similar responses based on recognized intent categories and question types. For simple enterprise-grade conversational AI experiences, a well-adopted approach is building a chatbot based on an existing comprehensive framework. Franco et al. (2020) embedded a RASA-based chatbot into Telegram software and trained an LSTM model to select a response for cybersecurity-related queries. Shukla et al. (2020) used the Hybrid Code Network (HCN), an RNN-based improved artifact proposed by Williams et al. (2017), to build a scoring model integrated into a chatbot designed in Microsoft Bot Framework SDK.

In the financial area, a trend in chatbot adoption over the past two years has been to migrate a mature development framework directly into specific applications. Financial firms prefer stable techniques that can accurately provide the specified responses. For example, RASA and Facebook Messenger have been widely used in the primary banking business (Bhattacharyya et al. 2020; Lothritz et al. 2021; Yang et al. 2021b), stock information query (Jiao 2020), and financial investment services (Yu et al. 2021). Previously, self-established attempts without a building template could also be found in the banking industry (Oh et al. 2018), where the authors created a fully connected neural network to detect user queries from the banking service domain.

5.1.2 Pipeline architecture with generative methods

A chatbot implemented with generative methods can add an NLU component to form a pipeline structure, increasing the coherence and relevance of the generated response. In some cases, many corpora were available to train a generative model. However, the model might get lost in the massive dialogue data and fail to capture topic-related features, generating universal but meaningless replies. Accordingly, some researchers introduced the NLU design to extract domain-related knowledge and conducted experiments to inspect whether the knowledge could help enhance the thematic coherence and relevance between the generated response and dialogue contexts. Liao et al. (2018) used an LSTM model for intent identification and an HRED model for style preference extraction to emphasize users' fashion needs in the generative model. Ren et al. (2020) and Haihong et al. (2020) improved customer services similar to the NLU of retrieval-based chatbots; they utilized the slot and response token information to assist in the generation process. These attempts revolved around the dialogue contents, and the mechanism was extended to an emotion-infused improvement. Instead of focusing on topic enhancement, Chang and Hsing (2021) focused on producing emotional replies. They designed a hybrid model of CNN and LSTM to predict user emotions and incorporate them to optimize the training process of the generative model.

5.2 Chatbot in end-to-end architecture

An end-to-end architecture means the design artifact can process original input data and produce final output without following the paradigm of step-by-step knowledge extraction. The development of deep learning technology enables these distributed operations to be integrated into a hierarchical model for collaborative learning of text features. From a supervised learning perspective, all constituent parts of an end-to-end model are trained

jointly to learn the data features and update the model parameters. A dialogue system built in this structure can lessen cumbersome manual intervention in the knowledge construction process. It can be regarded as an ensemble model with NLG functionality. Generally, researchers would utilize sufficient corpora of a specific domain to train a deep learning model with a complex structure to capture the syntactic and semantic features of dialogues as fully as possible. Similarly, chatbots in end-to-end architecture can also be differentiated by the common retrieval- or generation-based methods, essentially the same as the techniques used in the NLG component of the pipeline-based design.

5.2.1 End-to-end architecture with retrieval methods

This category of chatbots is employed with retrieval methods described in Sects. 4.3.1 (response scoring model as a regression task) or 4.3.2 (response selection model as a classification task) to directly determine a predefined response based on word embedding of input utterances. It applies to small and medium-scale scenarios (Prasomphan 2019b) where the prepared responses are limited, with the infrequent maintenance work of corpora. The deep learning model is trained to learn hierarchical features of the relationships between the user queries and correspondent responses from the given dialogue data, and this training process will be repeated for the whole dialogue system once the predefined corpora are updated. Hence, the end-to-end structure of high complexity may increase the difficulty of model tuning as the data volume increases.

The approach of building a scoring model in an end-to-end chatbot has been widely attempted in e-commerce services (Qiu et al. 2018; Prasomphan 2019a, b; Song et al. 2020) and customer technical support (Yang et al. 2018; Hardalov et al. 2019; Tahami et al. 2020; Damani et al. 2020). They utilized deep learning techniques to produce matching or ranking scores for the query and response candidates. As for treating the response-producing process as a classification task, the mapping from the input utterances to the multidimensional output places higher requirements on the structural design of neural networks to parse the input text information as much as possible. Li et al. (2021) leveraged an RNN, CNN, and BERT hybrid to build a complex response-selecting model in e-commerce conversation to classify the selection action. In addition, an upgrade of the combination with DRL was introduced to improve the long-term interactive performance of the response classification model corresponding to this dimensionally fixed multi-output structure. Williams et al. (2017) optimized the HCN model with a policy gradient-based reinforcement learning to enable interaction with various users and improve the consumer experience of restaurant services.

5.2.2 End-to-end architecture with generative methods

The generation-based chatbots were commonly built in end-to-end architecture after the LSTM model and Seq2seq structure proposal. Their unique generative performance of literally generating a response based on the given utterance and contexts was widely sought after by researchers. With the theoretical potential of answering any query, this particular structure enables this kind of chatbot to be applied in scenarios covering a wide range of conversation topics.

As mentioned in Sect. 4.3.3, the typical Seq2seq-based model has been examined in e-commerce (Kandasamy et al. 2017; Chen et al. 2019; Lin et al. 2021b; Ma et al. 2018),

social media (Xu et al. 2017; Kushwaha and Kar 2020, 2021; Aleedy et al. 2019), technical support (Aalipour et al. 2018; Golchha et al. 2019), and other applications (Olabiyi et al. 2019; Prajwal et al. 2019; Kang and Lee 2019). The techniques present powerful computing and abstraction capabilities to generalize and extract valuable knowledge and text features from massive dialogue data automatically to solve the response-generating problem, allowing developers to avoid the uncertainty and heavy workload associated with artificial feature engineering. However, this model still suffers shortcomings in the contextual logic, correctness, or coherence of the generated response at the current stage. Section 4.3.3 also introduced various methods for improvement from the perspective, such as combining with hybrid neural network models (Lin et al. 2021b; Aleedy et al. 2019), optimizing the objective function (Kandasamy et al. 2017; Golchha et al. 2019), embedding personalized features (Olabiyi et al. 2019), and adding external memory (Kang and Lee 2019). In general, these methods are difficult to apply in scenarios where the accuracy and professionalism of responses are critical, but their potential to generate new responses deserves further exploration.

6 Future research directions

Currently, research on business chatbots is still in the developing stage. Most published papers discussed applying relatively mature deep learning technologies to develop business chatbots. More research related to emerging technologies should be explored further. From our perspective, research on deep learning utilization in business chatbots should be extended concerning three aspects: (1) new scenarios and emerging technologies; (2) human–computer interaction and usability analysis; and (3) meta-theory and design principles for chatbot development. A summary of the research opportunity is shown in Table 7.

Table 7 Summary of possible research directions and topics

Research direction	Possible research topic
New scenarios and emerging technologies	Applying chatbots to new areas, such as individualized customer care, live streaming, online collaboration, short video marketing, and farmers' market digitization
	Integrating emerging technologies into chatbot design, such as burgeoning deep learning branches and efficient Quantum computing
Human–computer interaction and usability analysis	Assessing the effects of adopting the technology on human experience and activities
	Examining the practical usability for improving chatbot design
Meta-theory and design principles	A framework or guideline for designing a practical chatbot systematically and thinking about the value reasonably
	Meta-theories, evaluation indicators, or design principles to guide and regulate artifact construction

6.1 New scenarios and emerging technologies

Chatbots should be deployed in more industries or businesses to automate customer services because user queries can be handled 24/7, thus increasing productivity and reducing the need for human labor and expenditure. Researchers or developers have designed and implemented chatbot functions adapted to specific domain requirements. However, most existing chatbots are built for e-commerce scenarios or technical support of products where the business value is not fully reflected. Applying chatbots to new areas may have unexpected effects. Even Eliza's creators did not realize that chatbots could be used beyond psychology and that we are currently in a deep learning boom. This exploration requires us to observe the tedious or complex human tasks that can be automated or partially replaced in daily life, especially under the impact of COVID-19. Thus, broadening chatbot adoption into new areas, such as individualized customer care, live streaming, short video marketing, online collaboration, farmers' market digitization, and so on, is well worth exploring.

We should also try to integrate more emerging technologies into chatbot design to examine the improvement and adaptiveness of the technique. Attempts to apply new technologies to chatbots are conducive to popularizing chatbot technology. A chatbot is an application closely related to the development of computer technology, while the chatbot technology involved in business research is often less cutting-edge. In addition to being unable to tap into the commercial value of burgeoning technologies in time, studies using relatively backward technologies have difficulties attracting the interest of researchers from other fields, which is not conducive to enhancing the influence of our discipline in this field. For example, as a branch of deep learning, the practical commercial value of the CapsNet technique has rarely been reported in the business chatbot literature, which is expected to achieve better performance than CNN.

Apart from the fascinating algorithms used for better response production and plentiful functionalities, chatbot performance also benefits considerably from ground-breaking physics and engineering technology similar to the computational applications of neural networks inspired by biology. Quantum technology application is a field where an exponential set of input information can be manipulated simultaneously (Chen and Zhang 2014) from a micro Quantum perspective. The Quantum computational system can deal with data with distinctively quantum properties of uncertainty and entanglement (Bennett and DiVincenzo 2000) and represent more data states with less space than conventional computation. A possible Quantum chatbot is expected to receive multiple contradictory inputs from the environment and still develop an excellent response given the conflicting inputs. For example, a Quantum chatbot can react accurately to complicated human expressions in a targeted manner if complex emotions in an utterance can be understood and captured effectively.

We believe that as computer technology matures and develops, chatbot behaviors will be closer to human beings. Chatbots are also expected to be the next significant technological leap in automatic conversational services (Suhaili et al. 2021). Consequently, we should keep pace with advanced technology associated with chatbots and try to adapt and compare them for business chatbot design.

6.2 Human–computer interaction and usability analysis

The chatbot is an IT artifact that engages in frequent human–computer interaction, and user feedback is integral to the interaction. Considering and studying users' behaviors or

psychological activities is necessary when applying and designing chatbots. However, deep learning models are particularly perplexing because of their black-box nature (Gehrmann et al. 2020), confining human understanding and technology adoption (Gaur et al. 2021). Although this article conducts a thorough review and systematic analysis of applications and characteristics of mainstream deep learning technologies, how the adoption of different network techniques in chatbot design affects human–computer interactions remains unclear.

Inspired by a famous design science research framework proposed by Hevner et al. (2004), we address chatbot research by building and evaluating the artifact and testing whether the employed deep learning technologies meet the identified business needs. At this stage, where deep learning has been widely utilized in laboratory-level chatbot design, few studies have examined its human-oriented effects, limiting its extension to practical business scenarios that require high certainty and stability. An observation is that the banking industry prefers a pipeline-based chatbot framework that can explicitly demonstrate the information flow in chatbot systems with a predictable response produced. Therefore, apart from the feasibility of new technology, assessing the effects of adopting the technology on human experience and activities and examining its practical usability for improving chatbot design are also necessary. Although lacking in examinations of deep learning under chatbot contexts, some interesting studies can contribute to our better understanding. Luo et al. (2021) summarized several chatbot studies of usability analysis from the perspective of task performance, user attitudes, user trust, and user adoption, which could be extended to deep learning perception in chatbot design to unfold the mechanism of how it affects human beings.

6.3 Meta-theory and design principles for advanced chatbot development

A framework or guideline for designing a practical chatbot systematically and identifying its reasonable value may be necessary due to the different extents of mastery of various disciplines and technologies. The major challenge for chatbot development is its resource-intensive nature in terms of skills, time, and user interactions (Jackson and Latham 2022). With the rise of powerful technology, such as new deep learning and reinforcement learning methods, considering the technique features, business needs, and human factors in IT artifact design to improve performance comprehensively has become possible. As modern chatbots and their building technologies have become more diverse, more meta-theories or design principles are necessary to guide and regulate artifact construction.

Some researchers have been committed to generalizing practical frameworks or approaches to guide the chatbot design in some commercial applications. Lai et al. (2019) designed a security control procedure for banking chatbots and analyzed the e-commerce security strategies to reduce the security risk and concretely protect customer data security and personal privacy. Albert et al. (2019) affirmed the powerful impact of deep learning and considered its incorporation into a robust step-by-step development approach to business interaction systems. Sperli (2021) proposed a framework to model different cultural objects into a unified data model to support tourists' self-services. Other researchers have shared far-sighted views following characteristics of different scenarios based on their observations of chatbot performance. Prabowo et al. (2018) compared the LSTM and simple RNN models in the context of customer service talks across business fields. They summarized their performances and offered suggestions from the perspective of the

response accuracy and consumed time. Khan et al. (2019) conducted a correlative analysis of advanced AI technologies. They discussed the applications and inexpediencies of intelligent assistants to provide insight into adapting chatbots for specified scenarios.

Chatbot measurement methods are also necessarily upgraded to support advanced chatbot testing. Przegalinska et al. (2019) proposed a new approach to track and evaluate the interaction performance of deep learning-supported chatbots to help build better social bots fitting business or commercial environments. Huang et al. (2021) designed an assessment framework that can be used to identify the adoption susceptibility of chatbot applications in the hospitality and tourism industries.

The above studies provide a theoretical reference for the construction of business chatbots and guide individuals and enterprises to consider the aspects or indicators in evaluating the chatbot design. It is conducive to various disciplines learning from each other on the focus and improvement direction of appropriate and effective chatbot construction.

7 Conclusion

We present a thorough literature review on the design and applications of business chatbots in several application domains. We are currently in the middle of a boom in applying state-of-the-art deep learning techniques to design intelligent and adaptive chatbots. We first introduce two chatbot architectures that have evolved before conducting the mapping study. One of the main contributions of our study is the systematic illustration of the deep learning methods applied to construct business chatbots in the literature. We elaborate on the seven classes of computational approaches for business chatbot development. Another contribution of our study is that we summarize the four main usages of deep learning and compare their performance in each use. We thoroughly discuss the mainstream technology usages from the perspectives of natural language pre-processing, NLU, NLG, and external knowledge enhancement. The third contribution of our study is that it provides a new framework to classify chatbot construction architectures according to their technical characteristics. In particular, we differentiate the traditional classification of retrieval- and generation-based chatbots in terms of the pipeline and end-to-end structures. Finally, we highlight three promising future research directions for business chatbot design and development and call for a more profound exploration of the commercial values of business chatbots.

Acknowledgements Our research work was partly supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project: CityU 11507219) and a grant from the City University of Hong Kong SRG (Project: 7005196).

Author contributions YZ, RYKL, and JX wrote the main manuscript text; YZ and RYKL prepared figures and tables; RYKL provided the funding. All authors reviewed and edited the manuscript.

Funding This study was funded by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project: CityU 11507219) and a grant from the City University of Hong Kong SRG (Project: 7005196).

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

Aalipour G, Kumar P, Aditham S, Nguyen T, Sood A (2018) Applications of sequence to sequence models for technical support automation. In: 2018 IEEE international conference on big data, pp 4861–4869

Adamopoulou E, Moussiades L (2020) Chatbots: history, technology, and applications. Mach Learn Appl 2:100006. https://doi.org/10.1016/j.mlwa.2020.100006

Albert PS, Singh B, Das AS, Ieee (2019) A robust methodology for building an artificial intelligent (AI) virtual assistant for payment processing. In: 2019 IEEE technology & engineering management conference (TEMSCON)

Aleedy M, Shaiba H, Bezbradica M (2019) Generating and analyzing chatbot responses using natural language processing. Int J Adv Comput Sci 10(9):60–68. https://doi.org/10.14569/ijacsa.2019.0100910

Arsovski S, Osipyan H, Oladele MI, Cheok AD (2019) Automatic knowledge extraction of any chatbot from conversation. Expert Syst Appl 137:343–348. https://doi.org/10.1016/j.eswa.2019.07.014

Augello A, Vassallo G, Gaglio S, Pilato G (2009) A semantic layer on semi-structured data sources for intuitive chatbots. Cisis 1:760. https://doi.org/10.1109/Cisis.2009.165

Bartl A, Spanakis G (2017) A retrieval-based dialogue system utilizing utterance and context embeddings. In: 2017 16th IEEE international conference on machine learning and applications (ICMLA), pp 1120–1125. https://doi.org/10.1109/ICMLA.2017.00011

Bellman R (1954) The theory of dynamic programming. Bull Am Math Soc 60(6):503–515

Bennett CH, DiVincenzo DP (2000) Quantum information and computation. Nature 404(6775):247–255. https://doi.org/10.1038/35005001

Bhathiya HS, Thayasivam U (2020) Meta learning for few-shot joint intent detection and slot-filling. In: Pervasive health: pervasive computing technologies for healthcare, pp 86–92. https://doi.org/10.1145/3409073.3409090

Bhattacharyya S, Ray S, Dey M (2020) Context-aware conversational agent for a closed domain task. Adv Intell Syst Comput. https://doi.org/10.1007/978-981-15-2188-1_24

Bocklisch T, Faulkner J, Pawlowski N, Nichol A (2017) Rasa: open source language understanding and dialogue management. arXiv:1712.05181

Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Ling 5:135–146

Brahma AK, Potluri P, Kanapaneni M, Prabhu S, Teki S (2021) Identification of food quality descriptors in customer chat conversations using named entity recognition. In: CODS-COMAD 2021: proceedings of the 3rd ACM india joint international conference on data science & management of data (8th ACM IKDD CODS & 26th COMAD), pp 257–261. https://doi.org/10.1145/3430984.3431041

Canas P, Griol D, Callejas Z (2021) Towards versatile conversations with data-driven dialog management and its integration in commercial platforms. J Comput Sci. https://doi.org/10.1016/j.jocs.2021.101443

Chang YC, Hsing YC (2021) Emotion-infused deep neural network for emotionally resonant conversation. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2021.107861

Chen CLP, Zhang CY (2014) Data-intensive applications, challenges, techniques and technologies: a survey on big data. Inf Sci 275:314–347. https://doi.org/10.1016/j.ins.2014.01.015

Chen Y-N, Celikyilmaz A, Hakkani-Tur D (2018) Deep learning for dialogue systems. In: Proceedings of the 27th international conference on computational linguistics: tutorial abstracts, pp 25–31

Chen SQ, Li CL, Ji F, Zhou W, Chen HQ (2019) Review-driven answer generation for product-related questions in e-commerce. In: Proceedings of the twelfth ACM international conference on web search and data mining (WSDM'19), pp 411–419. https://doi.org/10.1145/3289600.3290971

Chiu MC, Chuang KH (2021) Applying transfer learning to automate annotation in an omni-channel system: a case study of a shared kitchen platform. Int J Prod Res 59(24):7594–7609. https://doi.org/10.1080/00207543.2020.1868595

Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Association for computational linguistics, pp 1724–1734. https://doi.org/10.3115/v1/D14-1179

Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555

Damani S, Narahari KN, Chatterjee A, Gupta M, Agrawal P (2020) Optimized transformer models for faq answering. In: Advances in Knowledge Discovery and Data Mining, PAKDD 2020, PT I, pp 235–248. https://doi.org/10.1007/978-3-030-47426-3_19

Devlin J, Chang M-W, Lee K, Toutanova K (2019) Bert: pre-training of deep bidirectional transformers for language understanding. In: Association for Computational Linguistics, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423

Elman JL (1990) Finding structure in time. Cogn Sci 14(2):179–211. https://doi.org/10.1207/s15516709cog1402_1

Fedus W, Goodfellow I, Dai AM (2018) Maskgan: Better text generation via filling in the_____. arXiv:1801.07736

Ferrod R, Cena F, Di Caro L, Mana D, Simeoni RG (2021) Identifying users' domain expertise from dialogues. In: UMAP 2021: adjunct publication of the 29th ACM conference on user modeling, adaptation and personalization, pp 29–34. https://doi.org/10.1145/3450614.3461683

Franco MF, Rodrigues B, Scheid EJ, Jacobs A, Killer C, Granville LZ, Stiller B (2020) Secbot: a business-driven conversational agent for cybersecurity planning and management. In: 2020 16th international conference on network and service management (CNSM)

Fukushima K (1980) Neocognitron: a self-organizing neural network model for a mechanism of pattern-recognition unaffected by shift in position. Biol Cybern 36(4):193–202. https://doi.org/10.1007/Bf00344251

Gaur M, Faldu K, Sheth A (2021) Semantics of the black-box: can knowledge graphs help make deep learning systems more interpretable and explainable? IEEE Internet Comput 25(1):51–59. https://doi.org/10.1109/Mic.2020.3031769

Gehrmann S, Strobelt H, Kruger R, Pfister H, Rush AM (2020) Visual interaction with deep learning models through collaborative semantic inference. IEEE Trans vis Comput Graph 26(1):884–894. https://doi.org/10.1109/Tvcg.2019.2934595

Golchha H, Firdaus M, Ekbal A, Bhattacharyya P (2019) Courteously yours: inducing courteous behavior in customer care responses using reinforced pointer generator network. In: NAACL HLT 2019: 2019 conference of the north american chapter of the association for computational linguistics: human language technologies: proceedings of the conference, pp 851–860

Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Adv Neural Inf Process Syst 27:2672–2680

Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press

Gori M, Monfardini G, Scarselli F (2005) A new model for learning in graph domains. In: Proceedings 2005 IEEE international joint conference on neural networks, vol 722, pp 729–734. https://doi.org/10.1109/IJCNN.2005.1555942

Guo JX, Lu SD, Cai H, Zhang WN, Yu Y, Wang J (2018) Long text generation via adversarial training with leaked information. In: Thirty-second AAAI conference on artificial intelligence/thirtieth innovative applications of artificial intelligence conference/eighth AAAI symposium on educational advances in artificial intelligence, pp 5141–5148

Haihong E, Zhan ZC, Song MN (2020) Table-to-dialog: building dialog assistants to chat with people on behalf of you. IEEE Access 8:102313–102320. https://doi.org/10.1109/Access.2020.2998432

Hakkani-Tur D, Tur G, Celikyilmaz A, Chen YN, Gao JF, Deng L, Wang YY (2016) Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. Interspeech 402:715–719. https://doi.org/10.21437/Interspeech.2016-402

Hardalov M, Koychev I, Nakov P (2019) Machine reading comprehension for answer re-ranking in customer support chatbots. Information. https://doi.org/10.3390/info10030082

Hatua A, Nguyen TT, Sung AH (2019) Goal-oriented conversational system using transfer learning and attention mechanism. In: 2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON), pp 99–104

He YH, Tang Y (2021) A neural language understanding for dialogue state tracking. Knowl Sci Eng Manag PT I:542–552. https://doi.org/10.1007/978-3-030-82136-4_44

He S, Liu K, An W (2019) Learning to align question and answer utterances in customer service conversation with recurrent pointer networks. In: 33rd AAAI conference on artificial intelligence, AAAI 2019, 31st innovative applications of artificial intelligence conference, IAAI 2019 and the 9th AAAI symposium on educational advances in artificial intelligence, EAAI 2019, pp 134–141

Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. Mis Quart 28(1):75–105. https://doi.org/10.2307/25148625
Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527
Hirschberg J, Manning CD (2015) Advances in natural language processing. Science 349(6245):261–266. https://doi.org/10.1126/science.aaa8685
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
Huang A, Chao Y, Velasco ED, Bilgihan A, Wei W (2021) When artificial intelligence meets the hospitality and tourism industry: an assessment framework to inform theory and management. J Hosp Tour Insights. https://doi.org/10.1108/Jhti-01-2021-0021
Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate cortex. J Physiol 148(3):574–591. https://doi.org/10.1113/jphysiol.1959.sp006308
Hubel DH, Wiesel TN (1968) Receptive fields and functional architecture of monkey striate cortex. J Physiol 195(1):215–243. https://doi.org/10.1113/jphysiol.1968.sp008455
Jackson D, Latham A (2022) Talk to the ghost: the storybox methodology for faster development of storytelling chatbots. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2021.116223
Jiao AR (2020) An intelligent chatbot system based on entity extraction using rasa nlu and neural network. In: 2020 4th international conference on control engineering and artificial intelligence (CCEAI 2020). https://doi.org/10.1088/1742-6596/1487/1/012014
Johnson M (2009) How the statistical revolution changes (computational) linguistics. In: Proceedings of the EACL 2009 workshop on the interaction between linguistics and computational linguistics: virtuous, vicious or vacuous?, pp 3–11
Jonke AW, Volkwein JB (2018) From tweet to chatbot: content management as a core competency for the digital evolution. In: Linnhoff-Popien C, Schneider R, Zaddach M (eds) Digital marketplaces unleashed. Springer, Berlin, pp 275–285. https://doi.org/10.1007/978-3-662-49275-8_28
Jordan MI (1986) Serial order: a parallel distributed processing approach. Technical report, June 1985–March 1986
Kandasamy K, Bachrach Y, Tomioka R, Tarlow D, Carter D (2017) Batch policy gradient methods for improving neural conversation models. In: 5th international conference on learning representations, ICLR 2017: conference track proceedings
Kang D, Lee M (2019) Seq-dnc-seq: context aware dialog generation system through external memory. In: 2019 international joint conference on neural networks (IJCNN)
Khan MA, Tripathi A, Dixit A, Dixit M (2019) Correlative analysis and impact of intelligent virtual assistants on machine learning. In: 2019 11th international conference on computational intelligence and communication networks (CICN 2019), pp 133–139. https://doi.org/10.1109/CICN.2019.24
Kim Y (2014) Convolutional neural networks for sentence classification. In, Doha, Qatar, association for computational linguistics, pp 1746–1751. https://doi.org/10.3115/v1/D14-1181
Kulkarni A, Mehta K, Garg S, Bansal V, Rasiwasia N, Sengamedu SH, Acm (2019) Productqna: answering user questions on e-commerce product pages. In: Companion of the world wide web conference (WWW 2019), pp 354–360. https://doi.org/10.1145/3308560.3316597
Kushwaha AK, Kar AK (2020) Language model-driven chatbot for business to address marketing and selection of products. IFIP Adv Inf Commun Technol. https://doi.org/10.1007/978-3-030-64849-7_3
Kushwaha AK, Kar AK (2021) Markbot: a language model-driven chatbot for interactive marketing in post-modern world. Inf Syst Front. https://doi.org/10.1007/s10796-021-10184-y
Kusner MJ, Hernández-Lobato JM (2016) Gans for sequences of discrete elements with the gumbel-softmax distribution. arXiv:1611.04051
Lai ST, Leu FY, Lin JW (2019) A banking chatbot security control procedure for protecting user data security and privacy. Lect Note Data Eng 25:561–571. https://doi.org/10.1007/978-3-030-02613-4_50
LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444. https://doi.org/10.1038/nature14539
Lee HK, Lee JS, Keil M (2018) Using perspective-taking to de-escalate launch date commitment for products with known software defects. J Manage Inf Syst 35(4):1251–1276. https://doi.org/10.1080/07421222.2018.1523604
Li FL, Qiu MH, Chen HQ, Wang XW, Gao X, Huang J, Ren JW, Zhao ZZ, Zhao WP, Wang L, Jin GW, Chu W, Assoc Comp M (2017) Alime assist: an intelligent assistant for creating an innovative e-commerce experience. In: CIKM'17: proceedings of the 2017 ACM conference on information and knowledge management, pp 2495–2498. https://doi.org/10.1145/3132847.3133169

Li L, Li CL, Ji DH (2021) Deep context modeling for multi-turn response selection in dialogue systems. Inf Process Manag. https://doi.org/10.1016/j.ipm.2020.102415
Liao LZ, Zhou Y, Ma YS, Hong RC, Chua TS (2018) Knowledge-aware multimodal fashion chatbot. In: Proceedings of the 2018 ACM multimedia conference (MM'18), pp 1265–1266. https://doi.org/10.1145/3240508.3241399
Lin Y, Wang H, Chen JN, Wang T, Liu Y, Ji H, Liu Y, Natarajan P (2021a) Personalized entity resolution with dynamic heterogeneous knowledge graph representations. In: ECNLP 4: the Fourth Workshop on E-commerce and NLP, pp 38–48
Lin ZH, Cui SB, Li GD, Kang XM, Ji F, Li FL, Zhao ZZ, Chen HQ, Zhang Y (2021b) Predict-then-decide: a predictive approach for wait or answer task in dialogue systems. IEEE ACM Trans Audio Speech 29:3012–3024. https://doi.org/10.1109/Taslp.2021.3110145
Liu B, Lane I (2016) Attention-based recurrent neural network models for joint intent detection and slot filling. Interspeech 1352:685–689. https://doi.org/10.21437/Interspeech.2016-1352
Liu S, Liu S, Xu W (2020) Gated attentive convolutional network dialogue state tracker. In: ICASSP, IEEE international conference on acoustics, speech and signal processing: proceedings, pp 6174–6178. https://doi.org/10.1109/ICASSP40776.2020.9054225
Lokman AS, Zain JM (2010) One-match and all-match categories for keywords matching in chatbot. Am J Appl Sci 7(10):1406–1411. https://doi.org/10.3844/ajassp.2010.1406.1411
Lothritz C, Allix K, Lebichot B, Veiber L, Bissyandé TF, Klein J (2021) Comparing multilingual and multiple monolingual models for intent classification and slot filling. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol 12801 LNCS. https://doi.org/10.1007/978-3-030-80599-9_32
Luo B, Lau RYK, Li CP, Si YW (2021) A critical review of state-of-the-art chatbot designs and applications. Wires Data Min Knowl. https://doi.org/10.1002/widm.1434
Ma CH, Ping G, Xin X (2018) Personalized response generation for customer service agents. Adv Neural Netw ISNN 2018:476–483. https://doi.org/10.1007/978-3-319-92537-0_55
Majid R, Santoso HA (2021) Conversations sentiment and intent categorization using context rnn for emotion recognition. In: 2021 7th international conference on advanced computing and communication systems, ICACCS 2021, pp 46–50. https://doi.org/10.1109/ICACCS51430.2021.9441740
McCulloch WS, Pitts WH (2016) A logical calculus of the ideas immanent in nervous activity. Embodiments Mind 1:19–38
Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst 26:1–10
Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533. https://doi.org/10.1038/nature14236
Moirangthem DS, Lee M (2018) Chat discrimination for intelligent conversational agents with a hybrid cnn-lmtgru network. In: Representation learning for NLP, pp 30–40
Mufadhol M, Wibowo A, Santoso JT (2020) Digital marketing techniques for business intelligence systems use automated chatbot machine learning. PalArch's J Archaeol Egypt/Egyptol 17(7):6895–6906
Nangoy JG, Shabrina NH (2020) Analysis of chatbot-based image classification on social commerce line@ platform. In: Proceedings: 2020 7th NAFOSTED conference on information and computer science, NICS 2020, pp 232–237. https://doi.org/10.1109/NICS51282.2020.9335874
Nuruzzaman M, Hussain OK (2020) Intellibot: a dialogue-based chatbot for the insurance industry. Knowl-Based Syst. https://doi.org/10.1016/j.knosys.2020.105810
Oh KJ, Lee D, Park C, Choi HJ, Jeong YS, Hong S, Kwon S (2018) Out-of-domain detection method based on sentence distance for dialogue systems. In: 2018 IEEE international conference on big data and smart computing (BIGCOMP), pp 673–676. https://doi.org/10.1109/BigComp.2018.00123
Olabiyi OO, Khazane A, Mueller ET (2019) A persona-based multi-turn conversation model in an adversarial learning framework. In: Proceedings: 17th IEEE international conference on machine learning and applications, ICMLA 2018, pp 489–494. https://doi.org/10.1109/ICMLA.2018.00079
Otterlo MV, Wiering M (2012) Reinforcement learning and markov decision processes. In: Reinforcement learning. Springer, pp 3–42
Paul A, Latif AH, Adnan FA, Rahman RM (2019) Focused domain contextual ai chatbot framework for resource poor languages. J Inf Telecommun 3(2):248–269. https://doi.org/10.1080/24751839.2018.1558378
Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543

Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Association for Computational Linguistics, pp 2227–2237. https://doi.org/10.18653/v1/N18-1202

Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic mapping studies in software engineering. In: 12th international conference on evaluation and assessment in software engineering (EASE) 12, pp 1–10

Petersen K, Vakkalanka S, Kuzniarz L (2015) Guidelines for conducting systematic mapping studies in software engineering: an update. Inf Softw Technol 64:1–18. https://doi.org/10.1016/j.infsof.2015.03.007

Pilato G, Augello A, Vassallo G, Gagho S (2007) Sub-symbolic semantic layer in cyc for intuitive chat-bots. In: Icsc 2007: international conference on semantic computing, proceedings. https://doi.org/10.1109/Icsc.2007.37

Prabowo YD, Warnars HLHS, Budiharto W, Kistijantoro AI, Heryadi Y, Lukas (2018) Lstm and simple rnn comparison in the problem of sequence to sequence on conversation data using bahasa indonesia. In: 2018 Indonesian association for pattern recognition international conference (INAPR), pp 51–56

Pradana A, Sing GO, Kumar YJ (2017) Sambot: intelligent conversational bot for interactive marketing with consumer-centric approach. Int J Comput Inf Syst Ind Manag Appl 9:265–275

Prajwal SV, Mamatha G, Ravi P, Manoj D, Joisa SK (2019) Universal semantic web assistant based on sequence to sequence model and natural language understanding. In: Proceedings of the 2019 9th international conference on advances in computing and communication, ICACC 2019, pp 110–115. https://doi.org/10.1109/ICACC48162.2019.8986173

Prasomphan S (2019a) Improvement of chatbot in trading system for smes by using deep neural network. In: 2019 IEEE 4th international conference on cloud computing and big data analysis (ICCCBDA), pp 517–522

Prasomphan S (2019b) Using chatbot in trading system for small and medium enterprise (smes) by convolution neural network technique. In: PervasiveHealth: pervasive computing technologies for healthcare, pp 93–98. https://doi.org/10.1145/3341069.3341092

Przegalinska A, Ciechanowski L, Stroz A, Gloor P, Mazurek G (2019) In bot we trust: a new methodology of chatbot performance measures. Bus Horizons 62(6):785–797. https://doi.org/10.1016/j.bushor.2019.08.005

Qiu MH, Yang L, Ji F, Zhou W, Huang J, Chen HQ, Croft WB, Lin W (2018) Transfer learning for context-aware question matching in information-seeking conversations in e-commerce. In: Proceedings of the 56th annual meeting of the association for computational linguistics, Vol 2, pp 208–213

Quan T, Trinh T, Ngo D, Pham H, Hoang L, Hoang H, Thai T, Vo P, Pham D, Mai T (2018) Lead engagement by automated real estate chatbot. In: Proceedings of 2018 5th nafosted conference on information and computer science (NICS 2018), pp 357–359

Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training

Ren XH, Yin HZ, Chen T, Wang H, Hung NQV, Huang Z, Zhang XL (2020) Crsal: conversational recommender systems with adversarial learning. ACM Trans Inf Syst. https://doi.org/10.1145/3394592

Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386–408. https://doi.org/10.1037/h0042519

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536. https://doi.org/10.1038/323533a0

Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: Advances in neural information processing systems, vol 30 (NIPS 2017)

Sanchez-Diaz X, Ayala-Bastidas G, Fonseca-Ortiz P, Garrido L (2018) A knowledge-based methodology for building a conversational chatbot as an intelligent tutor. Lect Notes Artif Int 11289:165–175. https://doi.org/10.1007/978-3-030-04497-8_14

Sanchez-Lengeling B, Reif E, Pearce A, Wiltschko AB (2021) A gentle introduction to graph neural networks. Distill 6(9):e33

Sandu N, Gide E (2019) Adoption of ai-chatbots to enhance student learning experience in higher education in india. In: 2019 18th international conference on information technology based higher education and training

Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80. https://doi.org/10.1109/TNN.2008.2005605

Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Serban IV, Sordoni A, Bengio Y, Courville A, Pineau J (2016) Building end-to-end dialogue systems using generative hierarchical neural network models. In: Thirtieth AAAI conference on artificial intelligence, pp 3776–3783

Serban IV, Sankar C, Germain M, Zhang S, Lin Z, Subramanian S, Kim T, Pieper M, Chandar S, Ke NR, Rajeshwar S, de Brebisson A, Sotelo JMR, Suhubdy D, Michalski V, Nguyen A, Pineau J, Bengio Y (2017) A deep reinforcement learning chatbot. arXiv:1709.02349

Setiaji B, Wibowo FW (2016) Chatbot using a knowledge in database human-to-machine conversation modeling. Proc Int Conf Intell. https://doi.org/10.1109/Isms.2016.53

Shalyminov I, Sordoni A, Atkinson A, Schulz H (2020) Fast domain adaptation for goal-oriented dialogue using a hybrid generative-retrieval transformer. In: 2020 IEEE international conference on acoustics, speech, and signal processing, pp 8039–8043

Sheikh SA, Tiwari V, Singhal S (2019) Generative model chatbot for human resource using deep learning. In: 2019 international conference on data science and engineering, ICDSE 2019, pp 126–132. https://doi.org/10.1109/ICDSE47409.2019.8971795

Shukla S, Liden L, Shayandeh S, Kamal E, Li JC, Mazzola M, Park T, Peng BL, Gao JF (2020) Conversation learner - a machine teaching tool for building dialog managers for task-oriented dialog systems. In: 58th annual meeting of the association for computational linguistics (ACL 2020): system demonstrations, pp 343–349

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D (2016) Mastering the game of go with deep neural networks and tree search. Nature 529(7587):484. https://doi.org/10.1038/nature16961

Singh R, Patel H, Paste M, Mishra N, Shinde N (2018) Chatbot using tensorflow for small businesses. In: Proceedings of the 2018 second international conference on inventive communication and computational technologies (ICICCT), pp 1614–1619

Song SY, Wang C, Chen HQ, Chen H (2020) Tcnn: triple convolutional neural network models for retrieval-based question answering system in e-commerce. In: WWW'20: companion proceedings of the web conference 2020, pp 844–845. https://doi.org/10.1145/3366424.3382684

Sperli G (2021) A cultural heritage framework using a deep learning based chatbot for supporting tourist journey. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2021.115277

Steinbauer F, Kern R, Kroll M (2019) Chatbots assisting german business management applications. Adv Trends Artif Intell Theory Pract 11606:717–729. https://doi.org/10.1007/978-3-030-22999-3_61

Steinhoff L, Arli D, Weaven S, Kozlenkova IV (2019) Online relationship marketing. J Acad Mark Sci 47(3):369–393. https://doi.org/10.1007/s11747-018-0621-6

Suhaili SM, Salim N, Jambli MN (2021) Service chatbots: a systematic review. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2021.115461

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. Adv Neural Inf Process Syst 27:27

Sutton RS, McAllester D, Singh S, Mansour Y (2000) Policy gradient methods for reinforcement learning with function approximation. Adv Neural Inf Process Syst 12(12):1057–1063

Tahami AV, Ghajar K, Shakery A (2020) Distilling knowledge for fast retrieval-based chat-bots. In: Proceedings of the 43rd international ACM sigir conference on research and development in information retrieval (SIGIR '20), pp 2081–2084. https://doi.org/10.1145/3397271.3401296

Thomas NT (2016) An e-business chatbot using aiml and lsa. In: 2016 international conference on advances in computing, communications and informatics (ICACCI), pp 2740–2742

Tiwari A, Saha T, Saha S, Sengupta S, Maitra A, Ramnani R, Bhattacharyya P (2021) A dynamic goal adapted task oriented dialogue agent. PLoS ONE 16(4):e0249030. https://doi.org/10.1371/journal.pone.0249030

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems 30 (Nips 2017)

Wei YG, Sun B, Sun MC, Zhao CY, Ma PZ (2014) Chinese intelligent chat robot based on the aiml language. Int Conf Intell Hum Mach 2:367–370. https://doi.org/10.1109/Ihmsc.2014.96

Wen MH (2018) A conversational user interface for supporting individual and group decision-making in stock investment activities. In: Proceedings of 4th IEEE international conference on applied system innovation 2018 (IEEE ICASI 2018), pp 216–219

Weng JJ, Ahuja N, Huang TS (1993) Learning recognition and segmentation of 3-d objects from 2-d images. In: Fourth international conference on computer vision: proceedings, pp 121–128

Williams JD, Asadi K, Zweig G (2017) Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning. In: Proceedings of the 55th annual meeting of the association for computational linguistics (ACL 2017), Vol 1, pp 665–677. https://doi.org/10.18653/v1/P17-1062

Wu YR, Mao WQ, Feng J (2021) Ai for online customer service: intent recognition and slot filling based on deep learning technology. Mobile Netw Appl. https://doi.org/10.1007/s11036-021-01795-5
Xu PY, Sarikaya R (2013) Convolutional neural network based triangular crf for joint intent detection and slot filling. In: 2013 IEEE workshop on automatic speech recognition and understanding (ASRU), pp 78–83
Xu AB, Liu Z, Guo YF, Sinha V, Akkiraju R (2017) A new chatbot for customer service on social media. In: Proceedings of the 2017 ACM sigchi conference on human factors in computing systems (CHI'17), pp 3506–3510. https://doi.org/10.1145/3025453.3025496
Xue Z, Ko TY, Yuchen N, Wu MKD, Hsieh CC (2019) Isa: Intuit smart agent, a neural-based agent-assist chatbot. In: IEEE international conference on data mining workshops, ICDMW, pp 1423–1428. https://doi.org/10.1109/ICDMW.2018.00202
Yang L, Qiu MH, Qu C, Guo JF, Zhang YF, Croft WB, Huang J, Chen HQ (2018) Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. ACM/SIGIR Proc 2018:245–254. https://doi.org/10.1145/3209978.3210011
Yang W, Xie Y, Lin A, Li X, Tan L, Xiong K, Li M, Lin J (2019) End-to-end open-domain question answering with bertserini. arXiv:1902.01718
Yang MH, Cao SS, Hu BB, Chen XL, Cui HB, Zhang ZQ, Zhou J, Li XL (2021a) Intellitag: an intelligent cloud customer service system based on tag recommendation. In: 2021 IEEE 37th international conference on data engineering (ICDE 2021), pp 2559–2570. https://doi.org/10.1109/ICDE51399.2021.00287
Yang YW, Hsu C, Tung HC, Shuai HH, Chang YJ (2021b) Tell me when users leave: predicting users' abandonment of a task-oriented chatbot service using explainable deep learning. In: ACM international conference proceeding series. https://doi.org/10.1145/3469595.3469630
Yu LT, Zhang WN, Wang J, Yu Y (2017) Seqgan: Sequence generative adversarial nets with policy gradient. In: Thirty-first AAAI conference on artificial intelligence, pp 2852–2858
Yu F, Zheng DQ, Zhao XT (2020) Multi-domain language understanding of task-oriented dialogue based on intent enhancement. In: 2020 international conference on asian language processing (IALP 2020), pp 221–228
Yu S, Chen YX, Zaidi H (2021) Ava: a financial service chatbot based on deep bidirectional transformers. Front Appl Math Stat. https://doi.org/10.3389/fams.2021.604842
Zhang Y, Wallace B (2015) A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv:1510.03820
Zhang X, Zhao JB, Yann LC (2015) Character-level convolutional networks for text classification. Adv Neural Inf 28:1–10
Zhang YZ, Gan Z, Carin L (2016) Generating text via adversarial training. In: NIPS workshop on adversarial training, academia. edu, pp 21–32
Zhang YZ, Gan Z, Fan K, Chen Z, Henao R, Shen DH, Carin L (2017) Adversarial feature matching for text generation. Proc Mach Learn Res 70:4006–4016
Zhang C, Li Y, Du N, Fan W, Yu PS (2018a) Joint slot filling and intent detection via capsule neural networks. arXiv:1812.09471
Zhang HN, Lan YY, Guo JF, Xu J, Cheng XQ (2018b) Tailored sequence to sequence models to different conversation scenarios. In: Proceedings of the 56th annual meeting of the association for computational linguistics (ACL), vol 1, pp 1479–1488
Zhang R, Wang ZY, Zheng MD, Zhao YY, Huang ZH (2021) Emotion-sensitive deep dyna-q learning for task-completion dialogue policy learning. Neurocomputing 459:122–130. https://doi.org/10.1016/j.neucom.2021.06.075
Zhao W, Ye J, Yang M, Lei Z, Zhang S, Zhao Z (2018) Investigating capsule networks with dynamic routing for text classification. In, Association for computational linguistics, pp 3110–3119. https://doi.org/10.18653/v1/D18-1350
Zhao GG, Zhao JY, Li Y, Alt C, Schwarzenberg R, Hennig L, Schaffer S, Schmeier S, Hu CJ, Xu FY (2019) Moli: smart conversation agent for mobile customer service. Information. https://doi.org/10.3390/info10020063
Zhao YY, Wang ZY, Yin K, Zhang R, Huang ZH, Wang P (2020) Dynamic reward-based dueling deep dyna-q: robust policy learning in noisy environments. In: Thirty-fourth AAAI conference on artificial intelligence, the thirty-second innovative applications of artificial intelligence conference and the tenth AAAI symposium on educational advances in artificial intelligence, pp 9676–9684

Zhou P, Shi W, Tian J, Qi ZY, Li BC, Hao HW, Xu B (2016) Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics (ACL 2016), vol 2, pp 207–212. https://doi.org/10.18653/v1/p16-2034

Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

Zhu Y, Janssen M, Wang R, Liu Y (2022) It is me, chatbot: working to address the covid-19 outbreak-related mental health issues in China: user experience, satisfaction, and influencing factors. Int J Hum Comput Interact 38(12):1182–1194. https://doi.org/10.1080/10447318.2021.1988236

Authors and Affiliations

Yongxiang Zhang[1] · Raymond Y. K. Lau[1] · Jingjun David Xu[1] · Yanghui Rao[2] · Yuefeng Li[3]

✉ Yongxiang Zhang
yongxiang.zhang@my.cityu.edu.hk

Raymond Y. K. Lau
raylau@cityu.edu.hk

Jingjun David Xu
davidxu@cityu.edu.hk

Yanghui Rao
raoyangh@mail.sysu.edu.cn

Yuefeng Li
y2.li@qut.edu.au

1 Department of Information Systems, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR

2 School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China

3 School of Computer Science, Queensland University of Technology, 2 George St, Brisbane City, QLD 4000, Australia

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com