International Journal of Automation and Computing 16(6), December 2019, 720-736
DOI: 10.1007/s11633-019-1200-0

Transfer Hierarchical Attention Network for Generative Dialog System

Xiang Zhang Qiang Yang

Computer Science and Engineering Department, Hong Kong University of Science and Technology, Hong Kong 999077, China

Abstract: In generative dialog systems, learning representations for the dialog context is a crucial step in generating high quality responses. The dialog systems are required to capture useful and compact information from mutually dependent sentences such that the generation process can effectively attend to the central semantics. Unfortunately, existing methods may not effectively identify importance distributions for each lower position when computing an upper level feature, which may lead to the loss of information critical to the constitution of the final context representations. To address this issue, we propose a transfer learning based method named transfer hierarchical attention network (THAN). The THAN model can leverage useful prior knowledge from two related auxiliary tasks, i.e., keyword extraction and sentence entailment, to facilitate the dialog representation learning for the main dialog generation task. During the transfer process, the syntactic structure and semantic relationship from the auxiliary tasks are distilled to enhance both the word-level and sentence-level attention mechanisms for the dialog system. Empirically, extensive experiments on the Twitter Dialog Corpus and the PERSONA-CHAT dataset demonstrate the effectiveness of the proposed THAN model compared with the state-of-the-art methods.

Keywords: Dialog system, transfer learning, deep learning, natural language processing (NLP), artificial intelligence.

1 Introduction

The Chit-chat dialog system is a promising natural language processing (NLP) technology which aims to enable computers to chat with human through natural language. Traditional chit-chat dialog systems are built by hand-crafted rules or directly selecting a human writing response from candidate pool using information retrieval (IR) technology[1–4]. These systems are not robust and it is difficult to deploy them in new domains. In recent years, deep learning has accomplished great success in various domains[5, 6] and a new paradigm called the generative dialog system achieves better performance than traditional works. The generative dialog system utilizes deep neural networks to model the complex dependency in dialog context and directly generate natural language utterances to converse with user. Several successful applications like Microsoft's XiaoIce[7] use generative dialog system technology and they are interacting with millions of people every day.

There are three basic components to build a generative dialog system: dialog context representation learning, response content selection and response generation. Given the dialog context, the model firstly learns a representation to encode the semantic information of the context. Then the model will decide the content for reply based on the dialog context representation. A final response will be generated by the language generation algorithm. By using a large scale human dialog corpus, all the three components are optimized jointly in an end-to-end paradigm to make the model emulate the agents in the training corpus[8].

Several methods have been developed for learning a representation of dialog context. In single turn dialog mode, the recurrent neural network (RNN) based encoder decoder model[9] is used. This sequence-to-sequence neural network is firstly proposed in the machine translation field and it beats the classical statistical machine translation (SMT) system. Given the current dialog sentence, an encoder RNN sequentially processes each word embedding and transforms the whole sentence into a fixed length vector. By training with backpropagation through time (BPTT), the fixed length vector is expected to encode necessary information of the input sentence for decoder to generate a response. Although this sequence-to-sequence model works well on short text conversation, RNN often forgets information passed through earlier states and thus the representation learning model is difficult to capture long term dependencies. To improve the inherent drawback, Shang et al.[10] propose to apply the attention neural network[11] in their sequence-to-sequence with attention dialog model. The context representation is the weighted sum of all states of the encoder RNN and this mechanism allows the model to pay

Research Article
Manuscript received June 27, 2019; accepted August 23, 2019; published online October 16, 2019
Recommended by Associate Editor Hong-Ji Yang

The original version of this article was revised due to a retrospective Open Access order

different attention to different words. The attention weights are dynamically learnt by a feed forward neural network in each decoding step. This model could represent the input sentence with less information loss. However, the attention mechanism suffers the problem that it is hard to attribute precise weights to crucial words and this will cause significant damage on the quality of generated response.

In multi-turn dialog mode, a dialog context usually contains several turns of interactions and the representation model needs to capture critical information and filter irrelevant words or sentences because the dialog state and the dialog topic may switch in each turn. The hierarchical recurrent neural network (HRED) adopts two RNN encoders in a hierarchical style to encode the dialog context[12]. The sentence level RNN is responsible for encoding each sentence into a high dimension vector which represents each sentence of the context. Another context level RNN is used to sequentially process each sentence representation vector and compress the whole dialog context into a fixed length vector. The dialog state changing and topic switching is captured through the state updating mechanism of context RNN during encoding. The problem of the HRED is that the context RNN easily forgets the previous dialog states. To address this issue, Xing et al.[13] propose the hierarchical recurrent attention network (HRAN) to learn a better representation of dialog context. In addition to two encoder RNN, a hierarchical attention mechanism is carefully designed to control the information contribution of each word and sentence into the final context representation. If a word or sentence is not related to current dialog state and topic, it should receive a low attention weight. However, the hierarchical attention mechanism also has the similar issue that its weight scores are suboptimal because they adopt a similar scheme as in [11]. The imprecise representation of dialog context will impact the quality of the downstream response generation module.

In this work, we aim to develop a more accurate dialog context representation model by proposing a novel attention mechanism. As we mentioned above, the bottleneck of the state-of-the-art representation model is the inaccurate attention scores. We assume the reason is because the information used to train the attention neural network is inadequate: the additive attention mechanism[11] just utilizes token information and the current decoder state to compute the weight score. Intuitively, it is trained in an unsupervised learning nature and the model does not have sufficient prior knowledge to identify crucial words and sentences in the dialog context. We think transfer learning is an effective approach to enhance the additive attention mechanism where keyword extraction and sentence entailment are used as auxiliary tasks to help the target model to obtain more reasonable weight scores. By transferring the knowledge of parsing syntactic structure and analyzing semantic relationships to target tasks, prior bias is injected and they are beneficial for determining important linguistic elements. This idea is also similar to recent advances in the machine translation field where word alignment information is used in attention networks to train it in a supervised learning style[14].

Based on the above motivation, we propose a novel transfer learning based attention mechanism and develop a new generative dialog framework: transfer hierarchical attention network (THAN). We apply two transfer learning methods to transfer knowledge from source task to target task: one is parameter pre-training and another one is network stacking. Various experiments have proved the effectiveness of these two methods[15–18]. We build a single-turn and a multi-turn dialog model based on the THAN and we conduct comprehensive experiments on large scale public datasets including quantitative evaluation and qualitative analysis. The results demonstrate that the THAN slightly outperforms the state-of-the-art models and it is able to generate logically consistent and semantically informative response.

The outline of the following sections is: In Section 2, we give a brief review of the related works in generative dialog systems, and we introduce the cutting-edge design of the attention mechanism. We also review the parameter pre-training and network stacking techniques of transfer learning which are applied in our work. The formal problem definition and the notations we used are introduced in Section 3. Then we give a detailed description of the models in Section 4 including the single-turn THAN, the multi-turn THAN and the auxiliary source task models. The experimental evaluations will be covered in Section 5 and we will discuss the conclusions and future directions in Section 6.

2 Related works

2.1 Generative dialog system

In the domain of chit-chat dialog systems, various generative models have been proposed. Some works focus on improving the performance of basic components like context representation and response generation. Researchers in [9, 10, 12, 13] use attention mechanisms and hierarchical RNN to learn the representation of dialog context, which is similar to the natural language understanding (NLU) module in task-oriented dialog systems[8]. Some works try to enhance the response generation module by using sophisticated generation algorithms[19–21]. To introduce more stochastic variability in generating high level structures like topics or goals of the response, VHRED[19] uses latent variables in the response generation phase and generates a response in a hierarchical way. Zhao et al.[20] propose to use conditional variational autoencoders (CVAE) in response generation to model the diversity of latent topics. Multiresolution recurrent neur-

al networks (MrRNN)[21] uses two parallel generation processes to generate the dialog response: The model firstly generates a high level concept sequence and then generates a natural language sentence based on the high level sequence.

In terms of training algorithms, reinforcement learning and adversarial training are adopted in addition to the supervised learning of minimizing the cross entropy between the generated response and the ground truth response. These learning schemes could help the dialog model to select more appropriate content. The reinforcement learning is able to improve the agent′s learning ability through interacting with external environments[22, 23]. Li et al.[24] propose a policy gradient method to reward the model when it selects content which could lead to an informative conversation in future turns. Their system takes the long term success of dialogues into consideration instead of simply choosing the response with maximum logarithm likelihood. The work in [25] utilizes the adversarial learning framework[26] to train a dialog model and a discriminator which distinguishes between the machine-generated and human-generated responses. The result from the discriminator is used as a reward to force the dialog model to emulate human agents.

On the other end of the spectrum, some works aim at adding additional features to generative dialog systems like boosting response diversity, keeping personality consistency and integrating external knowledge into the response. The state-of-the-art generative dialog models tend to generate safe but boring responses like "I don′t know". There are some works investigating this diversity inadequacy issue. The TA-Seq2Seq model[27] extracts the topic information in dialog contexts and incorporates them in response generation to produce informative and topic-aware responses. It adopts a joint attention mechanism to synthesize the message from a context hidden vector and topic vector, which is obtained from a LDA model[28]. The seq2BF model[29] firstly uses pointwise mutual information (PMI) to predict keywords from a dialog context and then generates responses by integrating the predicted keywords in a bi-directional way. It could produce responses which contains the keywords in appropriate positions. In terms of incorporating external knowledge into the dialog model, Zhou et al.[30] propose a method to utilize a large scale knowledge base to facilitate the dialog understanding context and generating a response. Knowledge graphs related to the input dialog context are retrieved from the knowledge base and encoded in a static representation. Then the model will dynamically attend to the retrieved knowledge graphs when generating the response to integrate external knowledge into conversation. Dialog agent persona consistency is another intriguing problem. Li et al.[31] propose a dual neural network to build a dialog system which acts like having consistent personality and background information. Speaker specific characteristics like speaking style are captured by a speaker model and the interaction pattern between two specific interlocutors is captured by a speaker addressee model. Those features will be used in response generation to facilitate the dialog model to keep a consistent personality.

2.2 Attention neural network

The attention neural network is an important structure which emulates the human cognitive process. It achieves great success in tasks like machine translation, image captioning and speech recognition. In addition to the original attention mechanism[11], Luong et al.[32] proposed to assemble it with the hard attention mechanism[33] to reduce the computation complexity. Both [11] and [32] are trained in an unsupervised learning setting where it is easy for the model to output sub-optimal weight scores. Liu et al.[14] propose a supervised attention mechanism for machine translation where the conventional alignment information between source and target sentence is used as supervision to train the network. Mi et al.[34] also propose a supervised attention mechanism by integrating the distance between machine attention and ground truth attention into the objective function and jointly optimizing it with the translation loss. In addition to the alignment information, other bias signals are introduced into attention mechanisms to guide it to output reasonable weight scores[35, 36]. Cohn et al.[35] incorporate structural bias like absolute positional bias, fertility, relative position bias and alignment consistency into the calculation of attention scores. Feng et al.[36] design a recurrent style attention mechanism to implicitly model the fertility and distortion of alignments between candidate and reference sentence. Their experiments demonstrate that the attention mechanism could be improved by adding prior bias in the machine translation task. To the best of our knowledge, there is no research trying to improve the attention mechanism in the generative dialog system.

2.3 Deep transfer learning in NLP

The goal of transfer learning is to transfer knowledge from a source model to a target model so the knowledge of the source model could be beneficial for the target model. As Pan and Yang point out[37], there are two types of transfer learning: 1) Transductive transfer learning is to transfer knowledge between different domain distributions where the source and target task are the same; 2) Inductive transfer learning aims at transferring knowledge between similar tasks. Our work could be categorized into the inductive transfer learning setting. With the prevalence of neural networks, deep transfer learning has revolutionized various fields like computer vision, speech recognition and natural language processing[38].

Parameter transfer is a widely adopted transfer learn-

ing approach which assumes the source task and target task share some parameters and the knowledge is transferred through the parameters[37]. In deep transfer learning, the parameters of the source task model are usually pre-trained and initialized on the target task model. Then the target task model will fine-tune the initialized parameters to adapt them to target domain distribution. The word embedding[39, 40] is an fundamental building block of many state-of-the-art deep learning NLP models. It is an empirically efficient transfer learning technique where the word vectors are trained on large scale open domain corpus and thus capture domain independent linguistic characteristics. Howard and Ruder[38] propose the FitLaM, a pre-trained language model whose parameters could be fine-tuned on any NLP tasks. FitLaM is a RNN based language model which could be generalized to different tasks by adding specific linear layers. Radford et al.[15] propose to use transformers[41] as the underlying language model architecture which consists of multi-layer self-attention to learn general sentence representation and outperforms RNN in many language understanding tasks[41–43]. Since RNN and transformers only model language token sequences in a unidirectional way, Devlin et al.[16] propose the bidirectional encoder representations for transformers (BERT) to make the pre-trained language model to learn representations which integrate both the left and the right context. By initializing the pre-trained parameters and fine-tuning them on target tasks, transfer learning achieves the state-of-the-art performance on eleven tasks[16].

Network stacking is also an effective approach to transfer knowledge between similar tasks where the output of the source task model is offered as input to the target task model. In deep neural networks, the knowledge of solving the source task is encoded in a dense vector and the target task model can obtain the knowledge by taking the dense vector into input. Such a hierarchical architecture is suitable for NLP tasks. Chen et al.[44] apply the stacking technique to jointly train two part-of-speech (POS) taggers on treebanks with different annotation standards so these two tasks could provide beneficial insights on sentence structures to each other. They propose two stacking schemes where the shallow stacking directly converts the source task′s predicted label into a embedding vector and feeds it to the target task and the deep stacking integrates the source task hidden feature vector into the input of the target task model. Søgaard and Goldbrg[17] transfer the knowledge from POS tagging to syntactic chunking and combinatory categorial grammer (CCG) supertagging by feeding the cascaded predictions to high level target tasks. The motivation is that POS tags are useful features for chunking and supertagging, e.g., the information that a word is a noun is useful in identifying the word′s syntactic chunk and its CCG supertag. Hashimoto et al.[18] propose a hierarchical network where five NLP tasks of different linguistic levels are stacked. Tasks in low levels like POS tagging, chunking and dependency parsing could provide syntactic and semantic information to high level tasks like semantic relatedness and sentence entailment. The hidden feature vector and the label embedding from low level task models are offered to high level tasks. Their experiments show that sophisticated tasks involving reasoning about the logical relationships of sentences could be beneficial for tasks aiming at analyzing morphology, syntax and semantics information.

3 Problem formalization and notations

The task of the generative dialog system is defined as following. Let Ω denote a dialog context and $\Omega = \{u_i\}_{i=1}^{m}$, where u_i is the i-th utterance and m is the number of utterances. We have $m = 1$ in single turn mode and in multi turn mode, we require $m \geq 2$. Let $u_i = \{w_{i,j}\}_{j=1}^{n_i}$ and $w_{i,j}$ is the j-th word in sentence u_i and n_i is the length of sentence u_i. The dataset of training corpus is $\mathcal{D} = \{(\Omega_i, Y_i)\}_{i=1}^{N}$, where $Y_i = \{y_{i,j}\}_{j=1}^{l_i}$ is the response to context Ω_i and $y_{i,j}$ is the j-th word in response Y_i. In the training phase, the dialog model aims to estimate the conditional distribution of $P(Y|\Omega)$ from dataset $\mathcal{D}$. In the inference phase, the model generates one response for each given context according to $P(Y|\Omega)$ and thus achieves conversation with the user.

In the dialog context representation learning phase, the model needs to compute an embedding vector $\boldsymbol{c}$ to encode the essential information of input context Ω. The representation $\boldsymbol{c}$ will be used in generation component to generate a sentence in response to context Ω.

In this paper, we use lower-case letters like a, b, c to represent scalars and bold letter case like $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$ to represent vectors. Upper-case letters in bold like $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ represent matrices.

4 Model design

4.1 Overview

The architecture of the THAN is shown as Fig. 1. It has two levels of RNN encoder and attention mechanism. From the bottom to top, each token of the input context is projected into an embedding space. The word level encoder transforms each word embedding into a fixed length vector by incorporating the information of the local context. Then the word level attention module computes an importance score for each word and aggregates word representations into a sentence representation. The sentence level encoder encodes all sentence representations and a sentence level attention module will decide attention weights for each sentence. All the sentence embeddings are compressed into a final context representation and passed to a RNN decoder. The decoder will generate one word in each time step based on the context representation using a greedy search or beam search al-

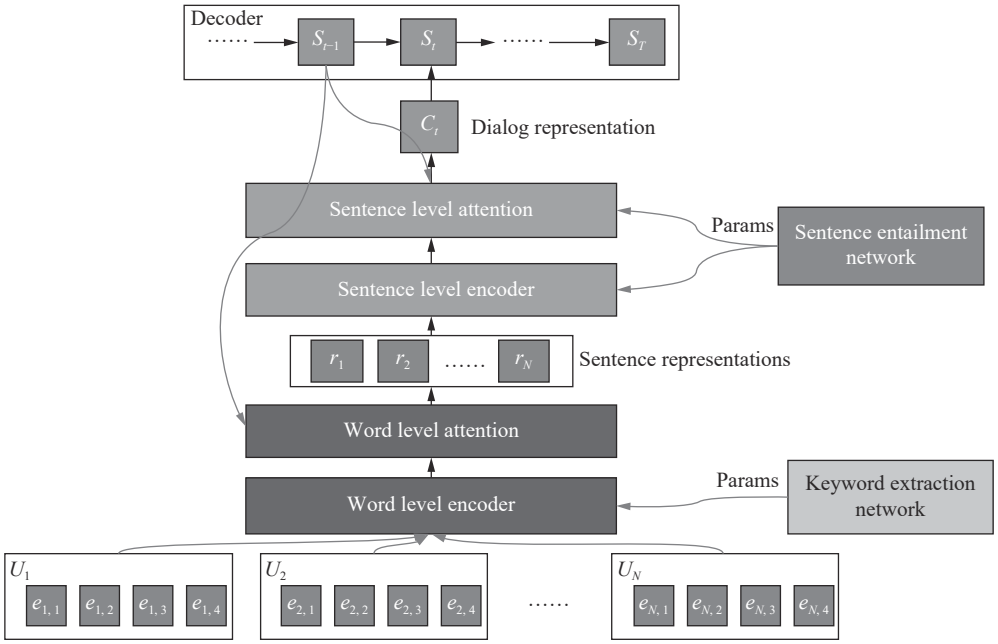

Fig. 1 Transfer hierarchical attention network

gorithm.

We will explain the motivation and methods we apply to conduct transfer learning in Sections 4.2. Sections 4.3 and 4.4 will describe the model for source tasks (auxiliary tasks). Then we will show the details of the THAN. Two generative dialog models are developed based on the THAN for single-turn dialog and multi-turn dialog respectively.

4.2 Transfer learning to THAN

The state-of-the-art attention models suffer the problem that they have relatively inadequate information to identify crucial linguistic elements in dialog context. In our hierarchical attention model, we use transfer learning to add prior knowledge to the target model. We select keyword extraction and sentence entailment as auxiliary source tasks and transfer the knowledge of parsing syntactic and semantic structure to our target task model:

Keyword extraction. As Hashimoto et al.[18] prove in their work, tasks from different linguistic levels of morphology, syntax, semantics can benefit each other in solving their own job. This is because the knowledge of solving low level NLP tasks like POS tagging can help to solve high level tasks like dependency parsing. The goal of source task word attention mechanism is to identify important words in a sentence and help the model to capture the sentence meaning. In order to achieve this, it needs to analyze the local syntactic and semantic information of each word. On the other side, a well-trained keyword extraction model is good at analyzing the meaning and dependencies of words to extract key phrases which summarize the central meaning of the sentence. By transfer learning from the keyword extraction task to word level attention mechanism, the model could represent each sentence more precisely by obtaining more accurate word weights.

Sentence entailment. In terms of sentence level attention, the target task model needs to detect topic continuation or switches to filter irrelevant information. In order to calculate the attention score of each sentence, sentence level attention should be able to decide whether a sentence is related to the current dialog topic or not. Intuitively, a less related sentence should be assigned a low score and vice versa. Therefore we transfer the knowledge of the sentence entailment model to the help target model to analyze the sentence relationships like entailment, neutral and contradiction. We believe this transfer learning based hierarchical attention component could enhance the dialog context representation learning and thus improve the final quality of generated responses.

We adopt two schemes to conduct transfer learning from source tasks (keyword extraction and sentence entailment) to target task (dialog context embedding).

1) Network stacking: In addition to the traditional token embedding vector and decoder state vector, we add the output feature vector of keyword extraction model as input to the word level attention network. The feature vector could be regarded as compressing the feature information of whether to classify a word as keyword and the word level attention module could calculate attention score based on these features. Similarly, the sentence level attention utilizes the output feature vector of the sentence entailment model when computing the sentence level importance score. This strategy is widely adopted in works on transfer learning from relatively straightforward tasks to sophisticated tasks, where one model′s output is used as input to another model[18, 44, 45].

2) Parameter pre-training: In addition to network

stacking, part of the parameters of word level encoder, sentence encoder and sentence level attention share the same structure and dimension as the parameter of source task models. We firstly pre-train the source task models and use the parameters to initialize the dialog model and then fine-tune them on the target task dataset, which is shown as a grey and yellow line in Fig. 1. As demonstrated by existing works on transfer learning, pre-training parameters on auxiliary source tasks and fine-tuning could further improve the target task model's performance[16].

4.3 Model for keywords extraction

The task of the keyword extraction model is to determine the words which could summarize the semantic information of the input sentence[46]. We transform the task to a supervised sequential prediction problem and train the model on a human annotated corpus. Fig. 2 shows the model. Given an input utterance of $\{w_i\}_{i=1}^m$, where w_i is the i-th token, an embedding lookup layer firstly projects each token into a pre-trained word embedding: $emb(\{\boldsymbol{w}_i\}_{i=1}^m) = \{\boldsymbol{e}_i\}_{i=1}^m$, where $\boldsymbol{e}_i$ is the embedding vector for word w_i. Then we adopt a Bi-directional RNN network with LSTM unit (Bi-LSTM) to encode each word embedding with its forward and backward context information[47, 48]. For each time step t, the input $\boldsymbol{g}_t$ for forward LSTM is defined as $\boldsymbol{g}_t = [\boldsymbol{h}_{t-1} : \boldsymbol{e}_t]$, the concatenation of word embedding and hidden state of last time step. The forward hidden state at time t is calculated as :

$$
\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_i \boldsymbol{g}_t + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_f \boldsymbol{g}_t + \boldsymbol{b}_f) \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_o \boldsymbol{g}_t + \boldsymbol{b}_o) \\
\boldsymbol{u}_t &= \tanh(\boldsymbol{W}_u \boldsymbol{g}_t + \boldsymbol{b}_u) \\
\boldsymbol{c}_t &= \boldsymbol{i}_t \odot \boldsymbol{u}_t + \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t)
\end{aligned}
$$

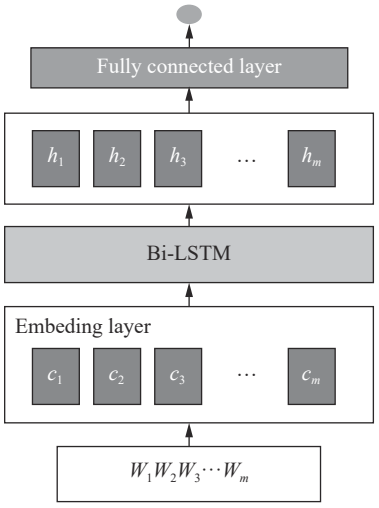

Fig. 2 Keyword extraction neural network

The $\boldsymbol{W}_i$, $\boldsymbol{W}_f$, $\boldsymbol{W}_o$, $\boldsymbol{W}_u$ are weight matrices and $\boldsymbol{b}_i$, $\boldsymbol{b}_f$, $\boldsymbol{b}_o$, $\boldsymbol{b}_u$ are bias vectors. σ is the sigmoid function and $\odot$ represents the element wise multiplication. The computation of the backward hidden state is conducted in reverse direction with a different set of parameters.

Each word is represented as the concatenation of its forward and backward hidden states. Then the word representations are passed into a one-layer fully-connected network with a hyperbolic tangent (tanh) activation function to obtain the feature vector. A softmax classifier is applied to calculate the probability $p(y_i|\boldsymbol{h}_i)$ of whether a specific word is a keyword. The objective function is to minimize the negative log-likelihood of the ground truth label of each word:

$$
J(\theta_{KE}) = -\sum_{d \in D} \sum_{i} \log p(y_i = \alpha | \boldsymbol{h}_i)
$$

where d is a sentence of dataset D and α is the correct label for the i-th word of d.

4.4 Model for sentence entailment

For a pair of natural language sentences (one is called premise and the other one is called hypothesis), the sentence entailment task is to classify whether the two sentences form a particular relationship like entailment, neutral or contradiction[49]. We use a sentence embedding model to solve this task and conduct supervised learning on human annotated corpus.

The model is shown in Fig. 3. For a pair of input sentences (s, s'), there are a couple of sentence embedding networks which encodes the "premise" and "hypothesis" respectively. Each sentence embedding module is a Bi-LSTM neural network and the whole sentence is represented as the concatenation of two hidden states: the final hidden state of forward LSTM and the final hidden state of backward LSTM. Then the representation of premise and hypothesis are concatenated into one feature vector which is fed into a three layer fully connected neural network. The activation function of the first two layers is rectified linear unit (ReLU) and a softmax function is

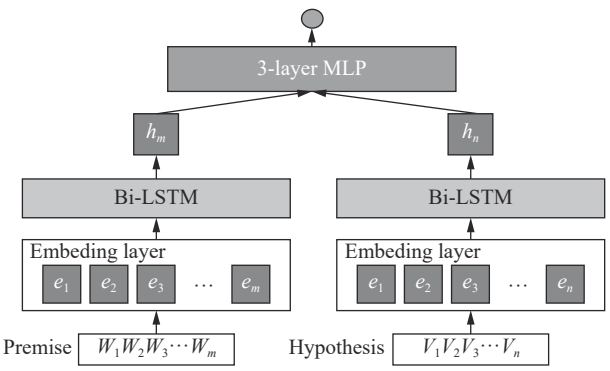

Fig. 3 Sentence entailment neural network

used in the last layer to output the probability vector of sentence relationship: $p(y_{s,s'}|\boldsymbol{h}_s, \boldsymbol{h}_{s'})$ where $\boldsymbol{h}_s$ and $\boldsymbol{h}_{s'}$ are sentence representations for premise and hypothesis respectively. The objective function is to minimize the negative log likelihood of ground truth label:

$$J(\theta_{SE}) = -\sum_{(s,s')} \log p(y_{s,s'} = \alpha|\boldsymbol{h}_s, \boldsymbol{h}_{s'})$$

where α is the correct label of sentence pair (s, s').

4.5 THAN for single-turn dialog

In the single-turn dialog mode, the model generates one response for one input sentence. The THAN could be applied to build a single-turn dialog system by using the word level encoder and word level attention mechanism to represent the context (only a single utterance in the case), then we pass the context embedding to the decoder to generate the response. The whole model is shown in Fig. 4.

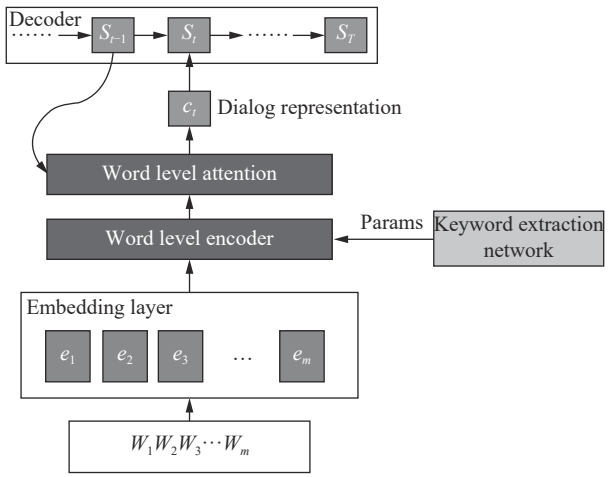

Fig. 4 Transfer hierarchical attention network for single-turn dialog

Given the input dialog context $\Omega = \{w_i\}_{i=1}^m$, an embedding lookup layer converts each token to a pre-trained word embedding. The word encoder is a Bi-LSTM neural network and it computes a sequence of hidden states $\{\boldsymbol{h}_i\}_{i=1}^m$ using the same formula as described in the keyword extraction model. Then each hidden state is fed into a fully-connected layer which has the same structure as the keyword extraction model's top layer in Fig. 4.2: $\boldsymbol{f}_i = \tanh(\boldsymbol{W}_{ke}\boldsymbol{h}_i + \boldsymbol{b}_{ke})$. The feature vectors $\{\boldsymbol{f}_i\}_{i=1}^m$ will be used in following attention mechanism.

The word level attention module dynamically calculates word weight scores in each decoding step. Assuming the decoder has generated t-1 words and the last decoder hidden state is $\boldsymbol{s}_{t-1}$, the importance score a_i for word w_i is defined by:

$$a_i = \frac{\exp(d_i)}{\sum_{k=1}^m \exp(d_k)} \quad d_i = \Theta(\boldsymbol{h}_i, \boldsymbol{s}_{t-1}, \boldsymbol{f}_i)$$

where $\boldsymbol{h}_i$ is the encoded hidden state and $\boldsymbol{f}_i$ is the keyword extraction feature vector computed from word level encoder for w_i. $\Theta()$ denotes a multi-layer fully-connected network with tanh activation funciton. Since the feature vector encodes the information of whether a particular word is crucial to represent the semantic information, the keyword extraction model could be thought of as giving a prior probability of attention score. The dialog model decides the final attention score based on current decoder state, which could be thought of as a posterior probability. In this way, the knowledge of the keyword extraction model is transferred to help the dialog model to calibrate the attention score but would not dominate it which may cause a negative transfer impact. The final representation of dialog context at time step t is defined as the weighted sum of word hidden states:

$$\boldsymbol{c}_t = \sum_{i=1}^m a_i \boldsymbol{h}_i \tag{1}$$

The decoder is a RNN language model[50] conditioned on previous generated words and dialog context Ω. The probability of generating the t-th word is defined by:

$$P(y_t|y_{t-1}, ..., y_1, \Omega) = \phi(e_{t-1}, \boldsymbol{c}_t, \boldsymbol{s}_t)$$

where $\{y_i\}_{i=1}^k$ is a set of random variables for generated words. $\phi()$ is a softmax function which calculates the next word probability distribution over the entire vocabulary. $\boldsymbol{s}_t$ is the decoder hidden state at time step t which is calculated as

$$\boldsymbol{s}_t = \mu(\boldsymbol{e}_{t-1}, \boldsymbol{s}_{t-1}, \boldsymbol{c}_t)$$

where μ denotes the LSTM unit as described before. The parameters of Bi-LSTM, $\boldsymbol{W}_{ke}$ and $\boldsymbol{b}_{ke}$ in word level encoder share the same structure and dimension with the keyword extraction model.

Let θ_{THAN} denote the parameter of THAN and we estimate θ_{THAN} from dataset $\{(\Omega_i, Y_i)\}_{i=1}^N$ by minimizing the negative log likelihood of ground truth response:

$$J(\theta_{THAN}) = \sum_{i=1}^N -\log(p(y_{i,1}, ..., y_{i,k_i}|\Omega_i)) + \delta||\theta_{share} - \theta'_{share}||^2$$

where θ_{share} is the shared parameters of the TAHN with keyword extraction model and θ'_{share} is the pretrained value of the shared parameters. The last regularization term is called "successive regularization term"[18] which can prevent the dialog model from forgetting the auxiliary knowledge from the source model.

4.6 THAN for multi-turn dialog

The input of a multi-turn dialog model is a sequence of utterances and the model generates responses conditioned on the input context. The model for multi-turn dialog is exactly the same as shown in Fig. 1. Given the context $\Omega = \{u_i\}_{i=1}^{n}$, where $u_i = \{w_{i,j}\}_{j=1}^{m_i}$ is the i-th utterance, we use the same embedding lookup layer and word encoder as described in the single turn model to compute word hidden states $\{\boldsymbol{h}_{i,j}\}_{j=1}^{m_i}$ before the decoding phase. Note that the word hidden states are calculated before decoding phase but the hierarchical attention mechanism and sentence level encoder will dynamically compute each sentence embedding as well as the whole context representation in each decoding step.

With last decoder state $\boldsymbol{s}_{t-1}$, the representation for the i-th sentence is defined as

$$\boldsymbol{r}_i = \sum_{j=1}^{m_i} a_{i,j}\boldsymbol{h}_{i,j}$$

where $a_{i,j}$ is the attention score for word $w_{i,j}$ and it is computed in the same way as in the single-turn dialog model: a multi-layer fully connected network with input as decoder state $\boldsymbol{s}_{t-1}$, word hidden state $\boldsymbol{h}_{i,j}$ and keyword extraction feature vector $\boldsymbol{f}_{i,j}$.

Since the sentence representation $\boldsymbol{r}_i$ only incorporates the information of local context in each sentence, we inject the dependency relation among sentences of the global dialog context by using a sentence level encoder, which is also a Bi-LSTM. For each sentence u_i, we use two LSTM units to compute its forward hidden state $\overrightarrow{\boldsymbol{x}_i}$ and backward hidden state $\overleftarrow{\boldsymbol{x}_i}$:

$$\overrightarrow{\boldsymbol{x}_i} = LSTM(\overrightarrow{\boldsymbol{x}_{i-1}}, \boldsymbol{r}_i), \quad \overleftarrow{\boldsymbol{x}_i} = LSTM(\overleftarrow{\boldsymbol{x}_{i+1}}, \boldsymbol{r}_i).$$

Then the final representation of each sentence is the concatenation of $\overrightarrow{\boldsymbol{x}_i}$ and $\overleftarrow{\boldsymbol{x}_i}$: $\boldsymbol{z}_i = [\overrightarrow{\boldsymbol{x}_i} : \overleftarrow{\boldsymbol{x}_i}]$. The sentence attention module computes each sentence attention weight by

$$b_i = \frac{\exp(d_i)}{\sum_{k=1}^{n}\exp(d_k)} \quad d_k = \Theta(\boldsymbol{z}_i, \boldsymbol{s}_{t-1}, \boldsymbol{g}_i)$$

where Θ is a multi-layer fully-connected network and $\boldsymbol{g}_i$ is the sentence entailment feature vector for sentence u_i and u_n, which will be discussed later. The final dialog context representation is the weighted sum of all sentence embeddings:

$$\boldsymbol{c}_t = \sum_{i=1}^{n} b_i \boldsymbol{z}_i \tag{2}$$

As we can see from equations 1 and 2, the two sets of attention weights controls how much contribution each hidden state makes to the final context representation: the more important a word or sentence is, the higher weight it will be assigned.

The sentence entailment feature vector is computed by the following formula: $\boldsymbol{g}_i = ReLU(\boldsymbol{W}_{SE}[\boldsymbol{z}_i : \boldsymbol{z}_n] + \boldsymbol{b}_{SE})$ where $\boldsymbol{z}_n$ is the hidden state of the last sentence in dialog context $\Omega = \{u_i\}_{i=1}^{n}$ and $\boldsymbol{W}_{SE}$ and $\boldsymbol{b}_{SE}$ are the shared parameters which have the same shape as the fully-connected layer in the sentence entailment model of Section 4.3. Since the last sentence is closely related to the current dialog state and topic, if $\boldsymbol{z}_i$ has close relation with $\boldsymbol{z}_n$, it has a high probability of relating to the current dialog state. The sentence entailment feature vector introduces the information of the relationship between sentences into sentence attention score learning and produces a prior probability of sentence importance. The dialog model computes the posterior attention weights based on the current generation state. Based on the context representation $\boldsymbol{c}_t$, the decoder works in a similar way as described before. The parameters of Bi-LSTM in sentence level encoder, $\boldsymbol{W}_{SE}$ and $\boldsymbol{b}_{SE}$ have the same shape as the sentence entailment task model, which will be initialized and fine-tuned in training phase. The objective function is similar to the single-turn THAN:

$$\begin{aligned} J(\theta_{THAN}) = &\sum_{i=1}^{N} -\log(p(y_{i,1}, ..., y_{i,k_i}|\Omega_i)) + \\ &\delta_1||\theta_{share-KE} - \theta'_{share-KE}||^2 + \\ &\delta_2||\theta_{share-SE} - \theta'_{share-SE}||^2 \end{aligned}$$

where $\theta_{share-KE}$ and $\theta_{share-SE}$ are the shared parameters of the TAHN with keyword extraction model and sentence entailment model respectively.

5 Evaluation

We will illustrate the experiments we conduct in this chapter. We compare the THAN with the state-of-the-art models by comprehensive evaluation including automatic test, human judgement, case study and data visualization.

5.1 Dataset

The dataset we use for single turn dialog response generation is the Twitter Dialogue Corpus[51]. The data is crawled from social media Twitter: User can post a query and his or her friends can reply to it. A single turn conversation consists of a query and a response. Table 1 shows some examples of the dataset. There are totally 851 963 dialogs. After removing the noisy data, it is spllitted into training, validation and test datasets, containing 689 537, 10 000, 1 000 dialogs respectively. The mean length of query and response is 13.8 words and 11.2 words respectively.

For the keyword extraction auxiliary task, the 500N-

KeyPhrasesCrowdAnnotated-Corpus is used[46]. The corpus contains 500 news articles from 10 diverse topics including sports, politics and entertainment which are similar as the topics covered in Twitter Dialog Corpus. Each topic contains 50 articles. Human annotators are recruited to label the key phrases in each sentence and the disagreement is solved by majority vote. Table 2 shows one example of the dataset. After pre-processing, 10 803 sentences are kept with mean length of 20.25 words. They are splited into training and validation dataset according to a ratio of 9:1.

The dataset for sentence entailment task is the Stanford Natural Language Inference corpus[49]. It collects pairs of sentence and human annotators categorize each of them into entailment, contradiction or neutral relationship. The dataset contains 570 152 sentence pairs which are two orders of magnitude larger than other datasets of this field. Table 3 shows some of the examples in SNLI corpus. We use the default dataset separation in their paper as 550 152 pairs for training, 10 000 pairs for validation and 10 000 pairs for testing. The mean length of premise and hypothesis is 14.071 words and 8.255 words respectively. The distribution of each label is illustrated in Table 4.

For the multi-turn dialog task, we use the PERSONA-CHAT dataset[52]. Comparing to other corpus in this field, this dataset is not crawled from open source websites by pre-defined rules like those used in [10, 13, 19], which are not real human dialogs and may include noisy data. PERSONA-CHAT is generated by real people on a crowd sourcing platform. Crowdworkers are randomly paired and required to converse naturally with each other on diverse topics including hobbies, movie and sports. This could produce various interesting and meaningful patterns which are valuable for model to emulate the agents in the corpus. The dataset contains 164 356 utterances which are larger than most of other corpus in this field. After pre-processing, we split the dataset into training, validation and testing with 131 438, 3 907 and 3 894 dialogs respectively. The mean number of turns is 14 and the mean length of utterance is around 11 words, which are extremely suitable to train multi-turn dialog model. One example of the dialog is shown in Table 5.

5.2 Baselines

We choose the following models as baseline:

Single-turn dialog: The sequence to sequence RNN encoder decoder model (S2S)[9] and its attention variation (S2SA)[10] are used as baselines in the single turn dialog task. Both of them are focusing on learning a representation for input sentence to enhance the final generation quality, which is consistent with our goals.

Multi-turn dialog: We compare the THAN with the HRED[12] and the HRAN[13] in multi-turn dialog setting. HRED uses RNN in both word and sentence level to encode the context and HRAN adopts hierarchical attention mechanism to control the contribution of each word and sentence in final representation. By comparing the performance of the THAN with these two baselines, we are able to investigate whether transfer learning from auxiliary tasks could improve the context representation or not.

5.3 Pre-processing and training procedure

To pre-process the 4 datasets, we use the open source natural language toolkit (NLTK) software to conduct

Table 1 Twitter dialog corpus

Query	Response
Someone really needs to pitch Landmark Theatres on a new site	Wow, that is literally the exact left half of my current browser window
I don't agree with everything he does! But I would never go out of my way to try and make him feel like crap!	In all honesty, he should feel like crap.

Table 2 500N-KeyPhrasesCrowdAnnotated-corpus

Text	Keywords
Big announcement today from Phish HQ: the quartet will be playing a three day festival in New York in July	announcement, quartet, three day festival

Table 3 Stanford natural language inference (snli) corpus

Premise	Hypothesis	Label
A boy is jumping on skateboard in the middle of a red bridge	The boy shows a skateboarding trick	entailment
A little league team try to catch a runner sliding into a base in an afternoon game	A team is playing baseball on saturn	contradiction
High fashion lady wait outside a tram beside a crowd of people in the city	The woman enjoy having a good fashion sense	neutral

Table 4 Distribution of labels in snli corpus

	Entailment	Neutral	Contradiction
Train	183416	182764	183187
Validation	3329	3235	3278
Test	3368	3219	3237

word segmentation and named entity recognition. All numbers were replaced by a special "[number]" token. We construct a vocabulary containing the most frequent 18423 words which accounts for 99% of total word appearance and the remaining words are replaced by a special "[unk]" token. We also append a "[eos]" token to represent the end of sentence after each utterance, which is suggested to be helpful for model to recognize the switch of dialog state[9].

We employ a two stage training procedure. The two source models are firstly trained until their performances converge on validation dataset. Subsequently, their parameters are used to initialize the corresponding shared part in the THAN and we fine-tune them on the target task. We select proper hyper-parameters like batch size, learning rate and LSTM cell dimensions to tune the auxiliary models until they achieve the same performance as reported in their source papers[46, 49]. We use the Glove vectors[40] to initialize the word embedding layer of all models. Other parameters are initialized by a Gaussian distribution with mean of 0 and standard deviation of 0.1.

We use Adam optimizer[53] to train the THAN and baseline models with an early stopping strategy on validation set. For each model, we use grid search to explore different hyper-parameter setting and choose the one with best performance to conduct evaluation. The hyper-parameter combinations we searched include: batch size of 32, 64, and 128; learning rate of 0.01, 0.001 and 0.0001; the dimensions of LSTM cell in word level, sentence level and decoder of 500, 600, 700. In order to avoid over-fitting, we apply gradient clipping and L2 regularization during the training phase. A beam search with width of 10 is adopted in inference phase and the top 1 response is used as final answer when calculating the performance metrics.

5.4 Quantitative evaluation

To examine the effectiveness of our context representation learning method in generative dialog system, we compare the model generated response with the ground truth response. However, the automatic evaluation of the response quality is still an open question because there is no single golden metric which strongly correlates with human judgements[54]. The human evaluation adopted in [9, 10] are proved to be more reliable than automatic evaluation. But it is difficult to be conducted in large scale evaluation and a small scale human evaluation may lack of statistical confidence. Therefore, we conduct automatic evaluation which includes multiple metrics and human evaluation to compare our model with baselines in a more all-round and fair way.

1) *Evaluation Metrics:* **Perplexity**. The first metric we use is word perplexity which is commonly used in probabilistic language model task[50, 55] and the baseline works[12, 13]. It is defined as

$$\exp\left(-\frac{1}{N_W}\sum_{i=1}^{N}\log P(Y_i|\Omega_i)\right)$$

where N is the size of testing dataset and N_W is the total number of words in dataset.

Intuitively, perplexity measures the model's ability to reconstruct the exact ground truth response and a lower perplexity indicates a better model. However, for a particular dialog context, there could be several possible re-

Table 5 Persona-chat dialog corpus

[P1:]	hi, how are you doing?
[P2:]	i'm getting ready to do some cheetah chasing to stay in shape.
[P1:]	you must be very fast. hunting is one of my favorite hobbies.
[P2:]	i am! for my hobby i like to do canning or some whittling.
[P1:]	that is neat. when i was in high school i placed 6th in 100 m dash!
[P2:]	that is awesome. do you have a favorite season or time of year?
[P1:]	i do not. but i do have a favorite meat since that's all i eat exclusively.
[P2:]	what is your favorite meat to eat?
[P1:]	i'd have to say its prime rib. do you have any favorite foods?
[P2:]	i like chicken or macaroni and cheese.
[P1:]	do you have anything planned for today? i think i am going to do some canning.
[P2:]	i am going to watch football. what are you canning?
[P1:]	i think i'll can some jam. do you also play footfall for fun?
[P2:]	if i have time outside of hunting and remodeling homes. which is not much!

sponses. So we also adopt three embedding based metrics: greedy matching, embedding average and vector extrema, all of which could approximate the sentence embedding by aggregating individual word embeddings[54]. The embedding based metrics are able to measure the semantic similarity between generated and ground truth response.

Greedy matching. Given the candidate response Y and the target response R, each token y in Y is greedily matched with the most similar token in R by cosine similarity between their word embeddings. The total score is the average similarity of all tokens:

$$G(Y,R)=\frac{\sum\limits_{y\in Y}\max\limits_{r\in R}cos\_sim(e_y,e_r)}{|Y|}.$$

The value $G(Y,R)$ is asymmetric and it is a common practice to compute the score in both directions and then obtain the average score $GM(Y,R)$:

$$GM(Y,R)=\frac{G(Y,R)+G(R,Y)}{2}.$$

Greedy match favours those candidate responses which have similar keywords as in target response.

Embedding average. The embedding average firstly calculates each sentence embedding by averaging corresponding elements in individual word vectors, which is widely adopted in textual similarity tasks[56]. The average embedding of a sentence S is calculated as

$$\bar{e}_S=\frac{\sum\limits_{w\in S}e_w}{|S|}.$$

The final score of embedding average is the cosine similarity between candidate response vector $\bar{e}_Y$ and ground truth vector $\bar{e}_R$.

Vector extrema. Vector extrema is an alternative way to calculate sentence embedding[57]. It takes the largest or the smallest value among all word embeddings in the sentence for each dimension. Let e_w^d denote the d-th dimension value of word vector e_w and the e_S denote the final embedding vector of sentence S. The d-th dimension value of e_S is defined as

$$e_S^d=\begin{cases}\max\limits_{w\in S}e_w^d, & \text{if} \quad e_w^d<|\min\limits_{v\in S}e_v^d|\\ \min\limits_{w\in S}e_w^d, & \text{otherwise.}\end{cases}$$

Given the embedding vector e_Y and e_R of candidate and target response, the score is also the cosine similarity. Because common words tend to appear in different context and they are pulled closer to the origin in embedding space. Vector Extrema is more likely to ignore common words and capture informative keywords by taking the extrema value among all dimensions.

Note that we don't use the word overlap similarity based metrics like bilingual evaluation understudy (BLEU), metric for evaluation of translation with explicit ordering (METEOR) and recall-oriented understudy for gisting evaluation (ROUGE) which are commonly adopted in machine translation task. They show weak correlation with human judgements according to the experiments in [54].

2) *Automatic Evaluation Results:* The evaluation results of single-turn dialog response generation is shown in Table 6, where S2S and S2SA stands for the sequence to sequence RNN encoder decoder model[9] and its attention version[10] respectively. THAN-KE is the single-turn version of our THAN model which conducts transfer learning from keyword extraction task as described in Section 4.4. According to Table 6, THAN achieves the lowest perplexity and the highest score on all three embedding based metrics than baselines, which shows THAN could generate high quality responses by transfer learning from keyword extraction task to word level attention mechanism.

Table 7 shows the results of multi-turn dialog re-

Table 6 Single-turn dialog response evaluation

	Perplexity	Greedy_Matching	Embedding_Average	Vector_Extrema
S2S	104.7	0.694	0.815	0.469
S2SA	102.5	0.723	0.839	0.481
THAN-KE	**101.2**	**0.746**	**0.857**	**0.530**

Table 7 Multi-turn dialog response evaluation

	Perplexity	Greedy_Matching	Embedding_Average	Vector_Extrema
HRED	39.318	0.616	0.748	0.527
HRAN	38.320	0.636	0.856	0.709
THAN-SE	37.221	**0.679**	0.878	0.717
THAN-KE-SE	**36.014**	0.645	**0.897**	**0.741**

sponse generation. We train two models of THAN with different auxiliary task setting. The THAN-KE-SE denotes the model with transfer learning from keyword extraction and sentence entailment task which is exactly the same as described in Section 4.5. THAN-SE is a naive variation which has the same architecture but only the sentence entailment task is transferred to the sentence level encoder and attention module. This variation is trained to verify the effectiveness of transfer learning from only sentence entailment task. As shown in Table 7, THAN-SE beats two baselines on all metrics, which demonstrates transfer learning from sentence entailment task can improve the context representation. THAN-KE-SE achieves best performance on perplexity, embedding average and vector extrema and comparable performance on greedy matching compared with THAN-SE. This shows that incorporating extra guidance in word level attention could further improve the final representation than only transfer learning to sentence level attention.

The score of THAN-KE-SE on greedy matching is slightly lower than the THAN-SE. We think the reason is because greedy matching favors those response whose words are semantically matching to keywords in ground truth response. THAN-KE-SE is better at extracting word level keywords and may tend to generate more informative response whose words are semantically relevant to the keywords in ground truth response. These responses are also logically reasonable response to the given context but may have a low greedy matching score. it also shows the necessity of adopting multi-criteria to evaluate a dialog model.

3) *Human Evaluation Results:* We also conduct human evaluation to compare the THAN with baselines in multi-turn dialog mode. Ten volunteers are hired and each of them annotates 60 different test cases. Each test case is annotated by 3 human volunteers and a majority vote strategy is adopted to decide the final result. Specifically, a dialog context with length of turns range from 5 to 15 is randomly drawn from test dataset and two responses were generated for it: one from the THAN and one from a baseline model. Each volunteer is firstly shown the dialog context and the two responses are then presented in random order. The human volunteer is required to choose a better response to answer the given context. The criteria is, response A is better than response B if A is relevant, logically consistent with given context but B is irrelevant, or logically contradictory to the context; or both responses are relevant with given context, but A is more informative than B. If the volunteer cannot tell which one is better, a "tie" label will be given. Totally 100 test cases are annotated for each "THAN VS baseline" pair. The results are presented in Table 8.

As we can see from the human evaluation results, the THAN outperforms both HRED and HRAN by winning more cases than loss in human judgements. This demonstrates that THAN is more likely to generate high relevant and informative response by encoding the dialog context in a better representation.

Table 8 Human evaluation results

	Win	Tie	Loss
THAN VS HRED	34	52	14
THAN VS HRAN	29	51	20

5.5 Qualitative evaluation

1) *Case Study:* We conduct case study to investigate the generated responses from the THAN and baseline models. Fig. 5 demonstrates some examples in single-turn dialog mode. We find that both the THAN and the S2SA can generate proper response to short queries like "Happy Birthday" and "I miss you". But in the case of long query setting like those in Fig. 5, S2SA usually generates safe response like "I see" (cases 1, 3 and 4) or logically inconsistent response (case 2). The analysis of attention visualization, which will be discussed in later section, shows that part of the reasons is because S2SA assigns inaccurate attention scores so it misses the critical information in the context. On the other hand, THAN is able to generate semantically relevant and informative response. In the first example, THAN captures that the query is talking about a sad experience and it tries to comfort the user. Taking the last case as another example, THAN correctly recognizes the central information of the query is about a TV drama and it outputs logically relevant response as "I haven't watched it yet".

In terms of multi-turn dialog mode, the ability for model to extract critical information from long turn conversation context is more important. As shown in the Fig. 6 of multi-turn dialog case study, it is hard for HRED to track the dialog state and current topic in long context, so HRED may generate logically contradictory response (case 2), or irrelevant response (case 1) or safe response (case 3). By using the hierarchical attention to model the word and sentence importance, HRAN partially alleviates the issue of forgetting long term context. For example, the context in case 2 is about a scenario where a people is late to an appointment and he is concern about his first impression, HRAN generates appropriate response of "I don't like it" to express the attitude towards the delay. However, HRAN generates information redundant response as in case 1, which has already been talked in previous turn. We find that HRAN suffers the similar issue of locating important sentences in relatively long context as in case 3, where it generates an irrelevant response. THAN performs better than HRAN in no matter short or long dialog context. In case 1, THAN "remembers" the previous topic of which part of California to live in and generate more precise response as "I am in North part". THAN also correctly captures the scen-

ID	Context	Response
1	In light of a work conversation i found a picture of my leg injury from 2006. I did not realize just how bad it really was at the time now.	S2SA: I see THAN-KE: I hope you feel better
2	People are still voting for Michael Lowry	S2SA: What do they do? THAN-KE: I think he is a bad leader.
3	All I know is you came to me when I was at my lowest. You picked me up breathed new life in me ... I owe my life to you	S2SA: You are great THAN-KE: Because I love you
4	Class is starting but now I can not focus because I'll be thinking about how Doctorwho is going to be at SDCC this year	S2SA: I like it THAN-KE: I haven't watched it yet

Fig. 5 Case study in single-turn dialog model

ID	Context	Response
1	U: Hey there. R: Hello! U: Where do you live? R: California and you? U: Same. R: North or south? U: I live in Irvine	HRED: What do you do for a living? HRAN: I'm in California THAN: I am in North part
2	U: Hi A: Hi, great, so why were you late? U: I was overslept. Does that bother you? A: What? yes! we need to be more on time U: I know, i know. Did 1 make a good first impression?	HRED: I am sure you are a good person HRAN: I do not like it THAN: I'm a bit of a guy who likes to say it is not good.
3	U: hi, what do you like to do? R: hi! I am very busy. i am next in line to be queen of England U: wow, impressive. i love to sing, even though i'm shy. R: www! that's fun! when i've free time i spend it with my cats U: i love cats! yeah, i have never been able to sing in public until lately. R: oh really? where? how did you decide to do it? U: i agreed to participate in an audition for a talent show.	HRED: That is great. HRAN: I like pets THAN: That is amazing! I hope you are successful

Fig. 6 Case study in multi-turn dialog model

ario in case 2 as HRAN does but generates more informative response to it. In case 3 where both HRED and HRAN fails to track the complex switch of dialog topics, THAN is able to identify the current dialog topic and generate logically consistent response. The case study demonstrates the potential of improving dialog context representation by rectifying the word and sentence weight score.

2) *Visualization of Attention:* In the analysis of visualization on word and sentence attention score, we can further prove the effectiveness of transfer learning from source task. Specifically, we mark the word or sentence with high attention score in different colors to examine which part of the context make a major contribution to the final context representation. Since the hierarchical attention mechanism computes a set of different weights in each decoding step, we average all of the weight scores obtained during the decoding phase for each word and sentence. Those with average score exceeding a threshold (the mean score of words or sentences in the context) is marked by colors in Figs. 7 and 8.

The visualization for cases of single turn dialog in Fig. 5 is illustrated in Fig. 7. Words with high attention score are marked as blue for S2SA and green for THAN in each line. The attention weights assigned by the THAN is generally more accurate than that of the S2SA. In the second case, S2SA predicts low attention weights on the "Michael Lowry" which is the object of verb "vote" and thus ignores it when generating response. But THAN gives a high attention score to it which matches our intuition and the response of THAN is directly related to the word "Michael Lowry". We feed the keyword extraction

ID	S2SA	THAN
1	In light of a work conversation i found a picture of my leg injury from 2006. I did not realize just how bad it really was at the time now	In light of a work conversation i found a picture of my leg injury from 2006. I did not realize just how bad it really was at the time now
	R: I see	R: I hope you feel better
2	People are still voting for Michael Lowry	People are still voting for Michael Lowry
	R: What do they do?	R: I think he is a bad leader
3	All I know is you came to me when I was at my lowest. You picked me up breathed new life in me ... I owe my life to you	All I know is you came to me when I was at my lowest. You picked me up breathed new life in me ... I owe my life to you
	R: You are great	R: Because I love you
4	Class is starting but now I can not focus because I'll be thinking about how Doctorwho is going to be at SDCC this year	Class is starting but now I can not focus because I'll be thinking about how Doctorwho is going to be at SDCC this year
	R: I like it	R: I haven't watched it yet

Fig. 7 Single-turn dialog attention visualization

feature vector of each word in case 2 into our keyword extraction model to obtain the keyword probability. The auxiliary model predicts high probability for words marked as green in Fig. 7, which means these words are classified as important keywords. We think the augmentation of attention score for word "Michael Lowry" is distilled from the knowledge transferred from source task model. It shows how target task leverages the prior bias to enhance context representation. In case 3, we observe that the word "breathed new life" is classified as keyword in auxiliary task but the dialog model does not give it a high attention score because it is not related to the central meaning of the whole context. This suggests that the design of our attention mechanism could prevent auxiliary model from dominating the prediction of dialog model, which may cause the negative transfer effect[37].

The attention weights for cases in Fig. 6 are presented in Fig. 8. Sentences with high score are marked orange or red in the left column and important words are marked as blue or green in each line. This graph also provides insights on how attention mechanism is improved in the THAN model compared to the HRAN model. In case 3, HRAN assigns high weights to the third and fourth sentences of the dialog context and it generates an utterance about pets in response to the content of the fourth sentence. This misleading attention score fails the HRAN to track the dialog state. THAN pays high attention in the last one and last third sentence and filters information which is not quite related to current topic. So THAN could generate response about participating the audition. Also, THAN "remembers" the sixth sentence in case 1 which is ignored by HRAN and THAN generates more

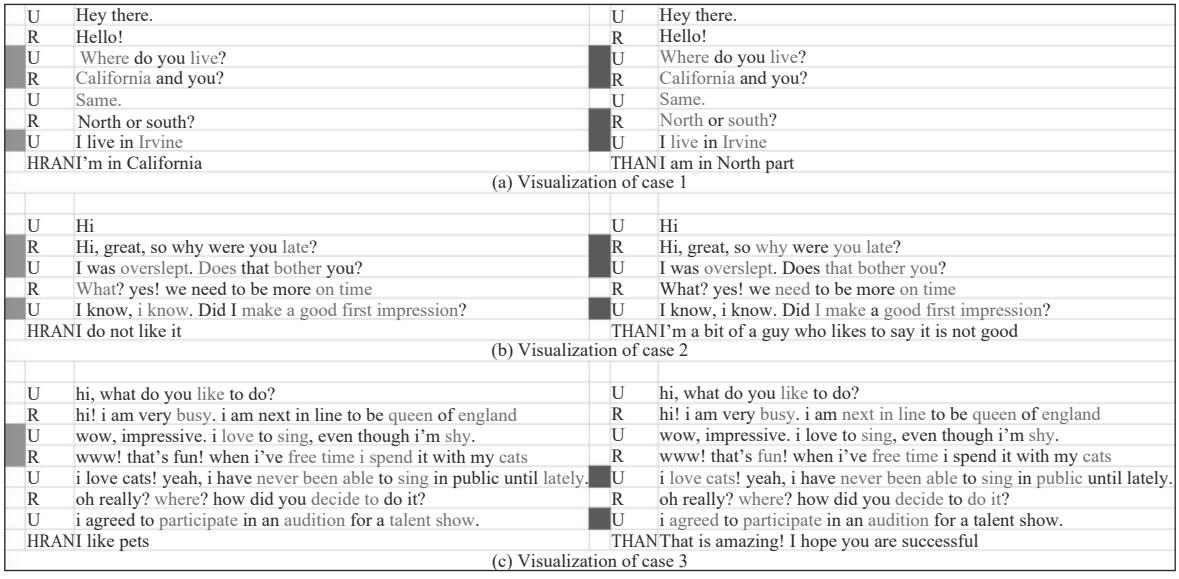

Fig. 8 Multi-turn dialog attention visualization

specific content in response to "where to live" than HRAN does. By transfer learning from sentence entailment task, THAN learns to analyze the sentence relationship and predicts more precise attention weights.

6 Conclusions

We attempt to develop an advanced generative dialog system by improving the context representation module. We propose a novel attention mechanism which uses transfer learning to predict precise attention scores and enhances the quality of response generation. The experiments show that the THAN model outperforms the baseline models. We can draw the following conclusions from our work:

1) Dialog context representation plays a crucial role in the generative dialog system and it deeply affects the final quality of generated responses. Representing context in an accurate formation could help the neural network to produce semantically relevant, logically consistent and informative response.

2) Transfer learning from keyword extraction and sentence entailment could provide useful prior knowledge to dialog model. It makes the model to learn the attention weights more precisely and thus more easily to extract essential information and track dialog states.

There are several future directions to extend. In addition to keyword extraction and sentence entailment task, we could consider conduct transfer learning from other NLP tasks like POS tagging, syntactic parsing and semantic relatedness. They are also fundamental language processing tasks and they can provide rich syntactic and semantic information to dialog model. Secondly, the current two auxiliary tasks are both trained by supervised learning whose performance may be limited by the amount of available data. It is worth to consider using unsupervised learning tasks like language model as auxiliary task. Moreover, we use a simple beam search decoder to generate the response and this may not be able to show the full potential of the context representation module. It would be intriguing to integrate the context representation module with more advanced generation model like reinforcement learning, GAN and conditional variational autoencoder to further improve the performance.

Open Access

References

[1] J. Weizenbaum. ELIZA – A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966. DOI: 10.1145/365153.365168.

[2] H. Wang, Z. D. Lu, H. Li, E. H. Chen. A dataset for research on short-text conversations. In *Proceedings of 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Seattle, USA, pp. 935–945, 2013.

[3] Y. Wu, W. Wu, C. Xing, Z. J. Li, M. Zhou. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Association for ComputationalLinguistics,Vancouver,Canada,pp. 496–505,2017.DOI: 10.18653/v1/P17-1046.

[4] X. Y. Zhou, D. X. Dong, H. Wu, S. Q. Zhao, D. H. Yu, H. Tian, X. Liu, R. Yan. Multi-view response selection for human-computer conversation. In *Proceedings of 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Austin, USA, pp. 372–381, 2016.

[5] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, Q. L. Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, vol. 14, no. 5, pp. 503–519, 2017. DOI: 10.1007/s11633-017-1054-2.

[6] Y. LeCun, Y. Bengio, G. Hinton. Deep learning. *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539.

[7] L. Zhou, J. F. Gao, D. Li, H. Y. Shum. The design and implementation of XiaoIce, an empathetic social chatbot. arXiv. preprint arXiv: 1812.08989, 2018.

[8] H. S. Chen, X. R. Liu, D. W. Yin, J. L. Tang. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 2, pp. 25–35, 2017. DOI: 10.1145/3166054.3166058.

[9] O. Vinyals, Q. V. Le. A neural conversational model. In *Proceedings of the 31st International Conference on Machine Learning Workshop*, Lille, France, 2015.

[10] L. F. Shang, Z. D. Lu, H. Li. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing,* Association for Computational Linguistics, Beijing, China, pp. 1577–1586, 2015.

[11] D. Bahdanau, K. Cho, Y. Bengio. Neural machine translation by jointly learning to align and translate, arXiv preprint, arXiv: 1409.0473, 2014.

[12] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, AAAI Press, Phoenix, USA, pp. 3776–3783, 2016.

[13] C. Xing, W. Wu, Y. Wu, M. Zhou, Y. L. Huang, W. Y. Ma. Hierarchical recurrent attention network for response generation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, AAAI Press, New Orleans, USA, 2018.

[14] L. M. Liu, M. Utiyama, A. Finch, E. Sumita. Neural machine translation with supervised attention. In *Proceed-*

ings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Association for Computational Linguistics, Osaka, Japan, pp. 3093–3102, 2016.

[15] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever. Improving language understanding by generative pre-training, [Online], Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf, 2018.

[16] J. Devlin, M. W. Chang, K. Lee, K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint, arXiv: 1810.04805, 2018.

[17] A. Søgaard, Y. Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Berlin, Germany, pp. 231–235, 2016.

[18] K. Hashimoto, C. M. Xiong, Y. Tsuruoka, R. Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, pp. 446–451, 2017.

[19] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, Y. Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI Press, San Francisco, USA, pp. 3295–3301, 2017.

[20] T. C. Zhao, R. Zhao, M. Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Vancouver, Canada, pp. 654–664, 2017. DOI: 10.18653/v1/P17-1061.

[21] I. V. Serban, T. Klinger, G. Tesauro, K. Talamadupula, B. W. Zhou, Y. Bengio, A. Courville. Multiresolution recurrent neural networks: An application to dialogue response generation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI Press, San Francisco, USA, pp. 3288–3294, 2017.

[22] M. Y. Zhang, G. H. Tian, C. C. Li, J. Gong. Learning to transform service instructions into actions with reinforcement learning and knowledge base. *International Journal of Automation and Computing*, vol. 15, no. 5, pp. 582–592, 2018. DOI: 10.1007/s11633-018-1128-9.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller. Playing Atari with deep reinforcement learning. arXiv preprint, arXiv: 1312.5602, 2013.

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

[25] J. W. Li, W. Monroe, T. L. Shi, S. Jean, A. Ritter, D. Jurafsky. Adversarial learning for neural dialogue generation. In *Proceedings of 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, pp. 2157–2169, 2017.

[26] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial nets. In *Proceedings of International Conference on Neural Information Processing Systems*, MIT Press, Montreal, Canada, pp. 2672–2680, 2014.

[27] C. Xing, W. Wu, Y. Wu, J. Liu, Y. L. Huang, M. Zhou, W. Y. Ma. Topic aware neural response generation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, AAAI Press, San Francisco, USA, pp. 3351–3357, 2017.

[28] D. M. Blei, A. Y. Ng, M. I. Jordan. Latent dirichlet allocation. *The Journal of machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[29] L. L. Mou, Y. P. Song, R. Yan, G. Li, L. Zhang, Z. Jin. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, Association for Computational Linguistics, Osaka, Japan, pp. 3349–3358, 2016.

[30] H. Zhou, T. Young, M. L. Huang, H. Z. Zhao, J. F. Xu, X. Y. Zhu. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI, Stockholm, Sweden, pp. 4623–4629, 2018.

[31] J. W. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. F. Gao, B. Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Berlin, Germany, pp. 994–1003, 2016.

[32] M. T. Luong, H. Pham, C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Lisbon, Portugal, pp. 1412–1421, 2015.

[33] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, pp. 2048–2057, 2015.

[34] H. T. Mi, Z. G. Wang, A. Ittycheriah. Supervised attentions for neural machine translation. In *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Austin, USA, pp. 2283–2288, 2016.

[35] T. Cohn, C. D. V. Hoang, E. Vymolova, K. S. Yao, C. Dyer, G. Haffari. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, San Diego, USA, pp. 876–885, 2016.

[36] S. Feng, S. J. Liu, M. Li, M. Zhou. Implicit distortion and fertility models for attention-based encoder-decoder NMT model. arXiv preprint, arXiv: 1601.03317, 2016.

[37] S. J. Pan, Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. DOI: 10.1109/TKDE.2009.191.

[38] J. Howard, S. Ruder. Fine-tuned language models for text classification. arXiv preprint, arXiv: 1801.06146, 2018.

[39] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International*

Conference on Neural Information Processing Systems, Curran Associates Inc., Lake Tahoe, USA, pp. 3111–3119, 2013.

[40] J. Pennington, R. Socher, C. D. Manning. GloVe: Global vectors for word representation. In *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543, 2014.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 5998–6008, 2017.

[42] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, N. Shazeer. Generating Wikipedia by summarizing long sequences. arXiv preprint, arXiv: 1801.10198, 2018.

[43] N. Kitaev, D. Klein. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Melbourne, USA, 2018.

[44] H. S. Chen, Y. Zhang, Q. Liu. Neural network for heterogeneous annotations. In *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Austin, USA, pp. 731–741, 2016.

[45] H. M. Wang, Y. Zhang, G. L. Chan, J. Yang, H. L. Chieu. Universal dependencies parsing for colloquial Singaporean English. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Vancouver, Canada, pp. 1732–1744, 2017. DOI: 10.18653/v1/P17-1159.

[46] L. Marujo, A. Gershman, J. Carbonell, R. Frederking, J. P. Neto. Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, European Language Resources Association, Istanbul, Turkey, 2012.

[47] A. Graves, J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005. DOI: 10.1016/j.neunet.2005.06.042.

[48] S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[49] S. R. Bowman, G. Angeli, C. Potts, C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Lisbon, Portugal, pp. 632–642, 2015.

[50] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, S. Khudanpur. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, ISCA, Makuhari, Chiba, Japan, pp. 1045–1048, 2010.

[51] A. Ritter, C. Cherry, W. B. Dolan. Data-driven response generation in social media. In *Proceedings of 2011 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Edinburgh, UK, pp. 583–593, 2011.

[52] S. Z. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, J. Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Melbourne, Australia, pp. 2204–2213, 2018. DOI: 10.18653/v1/P18-1205.

[53] D. P. Kingma, J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*, San Diego, USA, 2014.

[54] C. W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, J. Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Austin, USA, pp. 2122–2132, 2016.

[55] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[56] J. Wieting, M. Bansal, K. Gimpel, K. Livescu. Towards universal paraphrastic sentence embeddings. arXiv preprint, arXiv: 1511.08198, 2015.

[57] G. Forgues, J. Pineau, J. M. Larchevêque, R. Tremblay. Bootstrapping dialog systems with word embeddings. In *Proceedings of NIPS Workshop on Modern Machine Learning and Natural Language Processing Workshop*, Montreal, Canada, 2014.

Xiang Zhang is a master of Philosophy candidate of the Computer Science and Engineering Department in Hong Kong University of Science and Technology, China.

His research interests indude natural language processing, transfer learning and deep neural networks.

E-mail: xzhangax@cse.ust.hk (Corresponding author)

ORCID iD: 0000-0002-2822-5821

Qiang Yang received the Ph.D. degree form the University of Maryland, College Park, USA in 1989. He is the chief AI officer of WeBank, China′s first internet only bank with more than 100 million customers. He is also a chair professor at Computer Science and Engineering Department at Hong Kong University of Science and Technology, China. He is a Fellow of AAAI, ACM, IEEE, AAAS, and the founding Editor in Chief of the *ACM Transactions on Intelligent Systems and Technology* (ACM TIST) and the founding Editor in Chief of *IEEE Transactions on Big Data* (IEEE TBD). He has taught at the University of Waterloo and Simon Fraser University. He received the ACM SIGKDD Distinguished Service Award in 2017, AAAI Distinguished Applications Award in 2018, Best Paper Award of ACM TiiS in 2017, and the championship of ACM KDDCUP in 2004 and 2005. He is the current President of IJCAI (2017-2019) and an executive council member of AAAI.

His research interests include artificial intelligence, machine learning, especially transfer learning and federated machine learning.

E-mail: qyang@cse.ust.hk