Knowledge-Based Systems 243 (2022) 108485

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Increasing naturalness of human–machine dialogue: The users' choices inference of options in machine-raised questions

Xiaoling Zhou, Ou Wu *, Chao Jiang

National Center for Applied Mathematics, Tianjin University, Tianjin, 300072, China

ARTICLE INFO

Article history:
Received 30 August 2021
Received in revised form 17 February 2022
Accepted 19 February 2022
Available online 25 February 2022

Keywords:
Reverse QA
Human–machine dialogue
UCINet
SAGNet
Users' choices inference
Style-based text generation

ABSTRACT

In many practical applications, the machine needs to actively ask humans to obtain their intents. The process that the machine raises questions and users return answers is called reverse QA, which is an important part of a human–machine dialogue. However, in many dialogue systems, the machine restricts users from answering questions by clicking on option items, which is unnatural and restricted. In addition, this method may lose important information expressed by users. Users should be allowed to answer questions in natural language in a more natural and intelligent dialogue system. To obtain users' intents, users' choices of questions' options must be inferred from their answers. In this paper, we propose an advanced answer understanding network (UCINet) which infers users' choices of options in machine-raised questions accurately and efficiently according to the users' answer. Furthermore, metric learning is introduced for the model to learn better text representations. Based on the assumption that texts are determined by both semantics and styles, we propose a style-based answer generation network (SAGNet) which can generate various answers with different styles for a question. The generated answers are used to achieve data augmentation for UCINet's training. Experimental results on two reverse QA data sets demonstrate that UCINet achieves impressive results compared to other strong competitors. Using SAGNet for answer generation, we obtain answers with various styles and good quality. Our work can be widely used in intelligent customer service, mobile phone assistants, and other human–machine dialogue systems.

1. Introduction

In many human–machine dialogue systems, the machine needs to ask the user to obtain his/her accurate demand actively. That is, the machine raises a question that has several options, then the user answers it. Thereafter, the machine needs to understand the user's answer to obtain his/her intent accurately. This process is called reverse question–answering (QA) [1] which is one of the core parts of a human–machine dialogue. By reason of being able to obtain natural language responses, natural answers are more favored in real-world QA systems [2]. However, as shown in Fig. 1, in many commercial intelligent customer service systems, users are asked to give answers by clicking on the listed options. It not only brings a bad experience to users, but also it is unnatural and restricted. First, because each question has different options, users have to click one option after another to answer various questions in a human–machine dialogue. Obviously, this is exhausting and boring. Second, information may be missed in this way. It is not easy for the system to allow users to select multiple options. Thus, the user can only choose one option in most systems. If a user's intention is not among the options, the machine cannot give further feedback. In addition, users' neutral attitudes cannot be captured by the machine when users return answers by clicking on option items. The user's non-selection of an option does not necessarily mean the negation of the option. The user's attitude toward it may also be neutral. Therefore, it is inevitable to use natural language to answer questions in human–machine dialogue systems.

Accordingly, the machine should understand users' answers accurately and efficiently. In other words, users' choices of options in machine-raised questions must be inferred from their natural language answers, which significantly improves users' experience and increases the naturalness of the human–machine dialogue. Besides, more information about users' attitudes toward options can be obtained. Due to the richness of natural language and the diversity of users' answers, it is not a simple task to accurately infer users' choices of options from their answers. Table 1 shows the application of reverse QA for a complex task on booking a flight ticket.

It indicates that Reverse QA focuses on the machine's understanding of users' choices of options in machine-raised questions, which is a crucial part of the human–machine dialogue. Thus, it is

* Corresponding author.
E-mail addresses: xiaolingzhou@tju.edu.cn (X. Zhou), wuou@tju.edu.cn (O. Wu), jicii@126.com (C. Jiang).

https://doi.org/10.1016/j.knosys.2022.108485
0950-7051/

Table 1
The gray shading parts are the application scenarios of reverse QA. U and M are abbreviations of "User" and "Machine". Labels 0, 1, and 2 represent that the option is "un-chosen", "chosen" and "uncertain", respectively.

Role	Questions and answers	State
U	查询明天从北京去上海的机票。 Check out the ticket from Beijing to Shanghai tomorrow.	对话开始。 The conversation begins.
M	请问您想早上，下午还是晚上出发? Would you like to leave in the morning, afternoon or evening?	选择题 Multiple-choice question
U	早上我起不来。 I can't get up in time in the morning.	{早上、下午、晚上} 的标签为 {0,2,2}。 The label set of {Morning, afternoon, evening} is {0,2,2}.
M	请问您偏向于经济舱、商务舱还是头等舱? Would you like to buy business class, economy class or first class?	选择题 Multiple-choice question
U	都行。 They are all fine.	{经济舱、商务舱、头等舱} 的标签为 {1,1,1}。 The label set of {Economy Class, Business Class, First Class} is {1,1,1}.
M	以下是帮你查询到的机票信息，是否预定? The following is the ticket information for you to check, would you like to book it?	判断题 True/false question
U	可以，就订这个吧。 OK, I'll take it.	标签为1。 The label is 1.
M	已经帮您预订该航班机票，将跳转至付款页面！ The flight ticket has already been booked for you. Now we go to pay for the ticket!	任务完成。 Task completed.

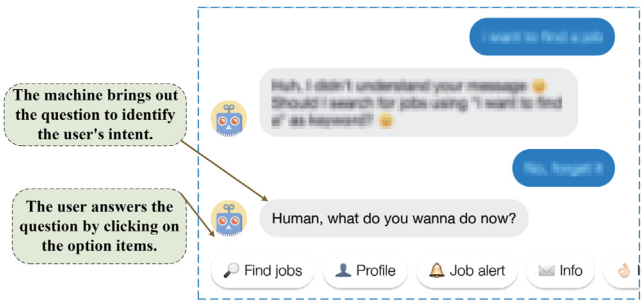

Fig. 1. An example of reverse QA in which the machine raises questions and users return answers by clicking on the listed options. This method for human–machine dialogue is especially unnatural and restricted.

urgent to improve the accuracy and efficiency of the reverse QA process, which is exactly our purpose. AntNet [1] was proposed inspired by reading comprehension, which can only predict a user's choice of one option at a time. If a question contains multiple options, the same user's answer must be input into the model multiple times, which is inefficient and ignores the relations among the user's choices of different options.

In this paper, we investigate the inference of users' choices of options from two aspects, namely, more effective users' choices inference and automatic answer generation (AG). First, we propose a new user choice inference network named UCINet that uses metric learning to get better text representations and processes all options in a question simultaneously. This network is more effective than AntNet and captures the correlation among a user's choices of different options. Second, inspired by question generation (QG) [3] for QA, AG is studied for reverse QA. We propose a style-based answer generation network (SAGNet) that integrates style types and can generate answers with multiple styles even under the same semantics.[1] The idea of the style-based answer generation can be integrated into most seq2seq models. The generated answers are added to the training data to achieve data augmentation [4].

The main contributions of our work are presented as follows:

- An effective user choice inference network (UCINet) is proposed, which can accurately obtain users' choices of options in machine-raised questions. Furthermore, metric learning is introduced for the model to learn better text representations. Experimental results show that UCINet significantly outperforms all other competitors, the accuracy of UCINet has increased by more than 8%, and the efficiency is improved compared with AntNet. The accuracy of UCINet with metric triplet loss has increased by 2%.
- Answer generation is investigated for the main task. SAGNet is proposed based on the style type for each answer which generates answers with different styles even under the same semantics. It improves the current situation where only a single answer can be generated for a question. It is proved that SAGNet can generate answers with diverse styles and good quality. Compared with the baseline, the BLEU-4 and ROUGE-L values have increased by 0.27 and 0.17. Answers generated by SAGNet are used to augment the training data. After that, the accuracies of UCINet with and w/o TL have increased by 1.75% and 2%.
- Our algorithm can be widely used in human–machine dialogue systems such as intelligent customer service, making them work more efficiently and naturally.

2. Related work

QA and reverse QA are reviewed at first. In QA, the machine aims to return a suitable answer to the user's question. The main method uses LSTM [5], CNN [6,7], or Transformer [8] to extract the features of questions and answers, and calculates the matching scores of the question and answer features, then returns the answer with the highest matching score to the user. Ref. [9] proposes a topic-enhanced LSTM model to improve document representations. ATICM [10] is used for automated answer type identification and classification by utilizing both syntactic and semantic analysis. MCAN [11] performs a series of soft attention operations, each time applying scalar features on internal word embeddings, which is used to sort sentences in dialogue modeling and question answering. HCAN [12] uses a hybrid encoder module based on CNN and LSTM. It also has a semantic matching module that captures contextual semantic relevance. To improve deep semantic matching, a method of focusing on keywords [13] is proposed. The measurement of similarity between questions and answers is important in QA systems. Thus, an effective similarity method for QA systems is proposed [14].

The difference between reverse QA and QA is that in reverse QA, the machine is the questioner, and the human is the answerer, which is the opposite of QA. At present, studies focusing on reverse QA remain limited. Semi-IAN [15] uses LSTM to model the semantic information of questions and answers, but less consideration is given to their interactive information. AntNet [1] considers the interactive information between questions and answers more effectively. It contains three key modules: unsupervised skeleton extraction for questions, relevance-aware answer representation, and multi-hop-based fusion. The disadvantages of AntNet are that only one option's label is inferred at a time, and the connection among users' choices of different options is ignored.

Our model is partially motivated by ABSA. Thus, ABSA is briefly reviewed here. Its purpose is to analyze the emotional tendency

[1] Answers in the same semantics mean that answers under the same question have the same choices of all options.

of every aspect involved in the text which is fine-grained sentiment analysis. ATAE [16] mainly uses the attention mechanism to capture the importance of contextual information to a given aspect. It combines the attention mechanism with LSTM to model the semantics of sentences. To capture long-distance emotional features, RAM [17] is proposed which is a multi-attention-based framework and is more robust to irrelevant information. These new models are mainly used to infer additional effective representations of sentences and aspects.

QG is a hot research issue in deep QA studies. It aims to generate questions to achieve data augmentation automatically, and the generated questions can be used to help the training of QA models [18]. Neural QG uses the seq2seq architecture [19] and the Transformer model [20] to generate questions. Ref. [21] proposes an end-to-end neural model for dialogue summarization with the supporting utterance flow modeling module and the fact regularization module. To distinguish the importance of different input words to the generated sentence, an attention-based sequence learning model is proposed [3] which is completely driven by data and does not require manual intervention. Recently, there are also some studies considering question types for QA via templates [22,23]. Motivated by QG, AG is investigated in our study, which can automatically generate answers to augment the training data set for reverse QA models.

3. Problem description

3.1. Problem definition

This paper follows the existing studies in which two types of representative questions are considered: true/false questions and multiple-choice questions. For true/false questions, because they have no options, the options of them are set to "None".

Given a machine question q_i containing a set of options o_i, o_i = $\{o_{i,1}, o_{i,2}, \ldots, o_{i,|o_i|}\}$ and a user's answer $a_{i,j}$, the users' choices inference task is to predict the labels on all options in a question. $|o_i|$ is the number of elements in the option set, that is, the number of options. Possible labels are 'un-chosen', 'chosen' and 'uncertain', represented by '0', '1', and '2', respectively. 'Uncertain' contains two situations, namely uncertain and irrelevant. Uncertain means that the user is unsure about this option, and irrelevant means that the user's answer does not mention this option.

In our study, we infer a user's choices of all options in a machine-raised question simultaneously, a training sample is (q_i, $a_{i,j}$, o_i, $l_{i,j}$) where q_i, $a_{i,j}$, and o_i represent the ith machine's question, the jth user's answer to the ith question, and the option set of the ith question, respectively. $l_{i,j}$ = $\{l_{i,j,1}, l_{i,j,2}, \ldots, l_{i,j,|o_i|}\}$ is the desired output of UCINet which indicates the user's choices of all options. $l_{i,j,k}$ is the label on the kth option of the jth answer which is a one-hot vector. For the learning of answer generation model, $a_{i,j}$ is the desired output. Taking a machine's question "Are you a boy or a girl?" and a user's answer "I am a girl". as an example, q_i is "Are you a boy or a girl?", $a_{i,j}$ is "I am a girl"., o_i = {"boy", "girl"}, and $l_{i,j}$ = {0, 1}.

3.2. Differences and connections with other tasks

Reverse QA is an emerging research problem that is different from previous tasks. The models in reverse QA aim to infer users' choices of options in machine-raised questions from users' natural language answers. It is worth noting that the users' choices inference task is different from the intent detection task [24,25]. Both of them are important sub-problems and should coexist in a human–machine dialogue system. The intent detection task is generally treated as a traditional multi-label classification problem whose label set is fixed in this case. Thus, the intent detection task mainly obtain the domain of the users' demands, but cannot directly capture the users' specific intents. It can be modeled using conventional classifiers including regression, support vector machines (SVMs), or even deep neural networks [25–27]. The special point is that this classifier is trained under a specific field. For example, all data in ATIS [28] data set is under the flight reservation domain with eighteen predefined labels. To predict the label of a sentence, the intent detection task commonly performs the following calculations after obtaining the sentence's representation h [25,29,30]:

$$l' = \text{softmax}\,(Wh + b)\,, \tag{1}$$

$$\mathcal{L} = -l \log l', \tag{2}$$

where l indicates the ground truth. Obviously, the users' choices inference requires more consideration of the interaction between questions and answers . In addition, the label set changes with the question in the users' choices inference. For example, in the question "Do you like literature, mathematics, or art?", the label set is {"literature", "mathematics", "art"}. But in the question "Are you a boy or a girl?", the label set is {"boy", "girl"}. There is no label set containing all options. Thus, it cannot be transformed into a multi-label classification task. However, models used for the feature extractor such as RNN [31], LSTM [5], and Transformer [8] are the same for both tasks.

The essence of our task is a bit close to Aspect-Based Sentiment Analysis (ABSA) [32] which aims to predict the emotional polarity of a given aspect or entity in the text. Therefore, some ideas of our model are borrowed from the ABSA models. However, the aspects and opinions of ABSA tasks are all extracted from the same sentence, while the aspects in reverse QA are all the options in a machine-raised question, and the opinions are users' choices of options which are extracted from users' answers.

4. Methodology

This section introduces two networks we proposed: UCINet for users' choices inference of options and SAGNet for automatic answer generation.

4.1. UCINet

This network infers a user's choices of all options in a machine-raised question simultaneously. The relationship among a user's choices of different options can thus be modeled. Using the idea of multi-task learning [33], metric learning is introduced to the model to get better text representations. The structure of UCINet is shown in Fig. 2, where the encoder of Transformer is used as the feature extractor [8].

4.1.1. Question and option encoding

This part is to model the information in questions and options. The input includes the word embeddings and rich feature vectors of questions. A position indicator vector is used to indicate options in the question.

$$\boldsymbol{h}_i{}^q = \text{Transformer\_encoder}\left(\boldsymbol{q}_i, \boldsymbol{I\_o}_i, \boldsymbol{A}_i^q\right), \tag{3}$$

where $\boldsymbol{h}_i^q$ is the Transformer representation of the ith question, $\boldsymbol{q}_i$ and $\boldsymbol{A}_i^q$ represent the word embedding and the rich feature vector of the ith question, the position indicator vector of the mth word in the ith question $\boldsymbol{I\_o}_{i,m}$ is $\mathbf{0}$ or $\mathbf{1}$ ($\mathbf{1}$ means that the word belongs to options' words, and $\mathbf{0}$ is the opposite). The question representation is $\boldsymbol{h}_i^q$ = $[\boldsymbol{h}_{i,1}^q, \ldots, \boldsymbol{h}_{i,m}^q, \ldots, \boldsymbol{h}_{i,M_i}^q]$ and $\boldsymbol{h}_{i,m}^q \in \mathbb{R}^d$. M_i is the length of the ith question.

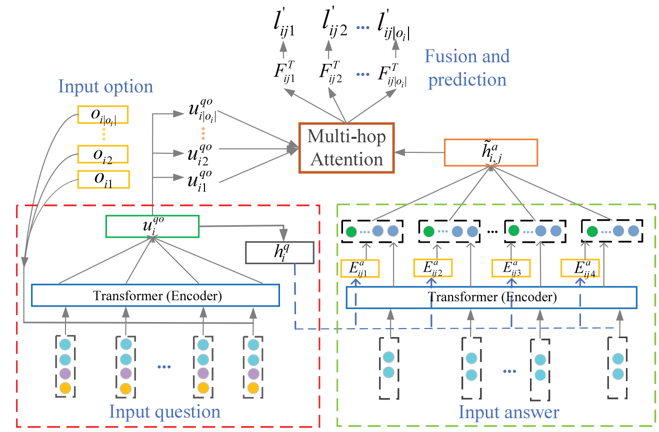

Fig. 2. Structure of UCINet. This network contains four main modules: question and option encoding, answer encoding, information fusion and prediction, and metric triplet loss.

To obtain the option representations, we input the options' embeddings into the encoder of Transformer. Let $\boldsymbol{h}^o_{i,k} = [\boldsymbol{h}^o_{i,k,1}, \boldsymbol{h}^o_{i,k,2}, \ldots, \boldsymbol{h}^o_{i,k,L_{ik}}]$ represent the kth option's representation and L_{ik} represent the length of the kth option after word segmentation.

The question representation $\boldsymbol{h}^q_i$ is concatenated with the representation of each option in the ith question separately. The concatenation vector of the question and the kth option is $\boldsymbol{h}^{qo}_{i,k} = [\boldsymbol{h}^q_i, \boldsymbol{h}^o_{i,k}] = [\boldsymbol{h}^{qo}_{i,k,1}, \ldots, \boldsymbol{h}^{qo}_{i,k,M_i+L_{ik}}]$. The final question and option representation is calculated on the basis of the attention score for each word which is calculated as follows:

$$\boldsymbol{\alpha}^{qo}_{i,m,k} = \left(\boldsymbol{h}^{qo}_{i,k}\boldsymbol{w}_a\right)^T \boldsymbol{h}^{\boldsymbol{q}}_{\boldsymbol{i},\boldsymbol{m}}, \tag{4}$$

$$\boldsymbol{att}^{qo}_{i,m,k} = \frac{exp\left(\boldsymbol{\alpha}^{qo}_{i,m,k}\right)}{\sum_{m=1}^{M_i} exp\left(\boldsymbol{\alpha}^{qo}_{i,m,k}\right)}, \tag{5}$$

where $\boldsymbol{w}_a \in \mathbb{R}^{M_i+L_{i,k}}$. The final question and option representation is

$$\boldsymbol{u}^{qo}_{i,k} = \sum_{m=1}^{M_i+L_{ik}} att^{qo}_{i,k,m}\boldsymbol{h}^{qo}_{i,k,m}. \tag{6}$$

Answer encoding

This part models the answer information. The input includes answer's word embeddings and rich feature vectors.

$$\boldsymbol{h}^a_{i,j} = \text{Transformer\_encoder}\left(\boldsymbol{a}_{i,j}, \boldsymbol{A}^a_{i,j}\right), \tag{7}$$

where $\boldsymbol{a}_{i,j}$ and $\boldsymbol{A}^a_{i,j}$ are the word embedding and rich feature vector of the jth answer. The answer representation is $\boldsymbol{h}^a_{i,j} = [\boldsymbol{h}^a_{i,j,1}, \ldots, \boldsymbol{h}^a_{i,j,2}, \ldots, \boldsymbol{h}^a_{i,j,N_{i,j}}]$ and $N_{i,j}$ is the answer's length. To measure the importance of each word in the answer relative to the question, a relevance score $\boldsymbol{E}^a_{i,j,n}$ is calculated whose calculation is the same as AntNet [1]. The answer's final representation is

$$\widetilde{\boldsymbol{h}}^a_{i,j,n} = \begin{bmatrix} \boldsymbol{h}^a_{i,j,n} \\ \boldsymbol{E}^a_{i,j,n} \end{bmatrix}, n = 1, \ldots, N_{i,j}, \tag{8}$$

where $\boldsymbol{h}^a_{i,j,n} \in \mathbb{R}^d$ is the Transformer representation of the nth word in the jth answer, $\boldsymbol{E}^a_{i,j,n} \in \mathbb{R}^{d_e}$ is the question–answer correlation reward vector.

4.1.2. Information fusion and prediction

The multi-hop attention mechanism is used to fuse the representation of answers, options, and questions. Let $\boldsymbol{F}^0_{i,j,k}$ represent the initial fusion vector.

$$\boldsymbol{F}^0_{i,j,k} = \boldsymbol{u}^{qo}_{i,k}. \tag{9}$$

The calculation of the first hop is presented as follows:

$$\boldsymbol{z}^1_{i,j,k,n} = \boldsymbol{W}^{1T}_z \tanh\left(\boldsymbol{w}^1_h\widetilde{\boldsymbol{h}}^a_{i,j,n} + \boldsymbol{w}^1_x\boldsymbol{F}^0_{i,j,k} + b^1\right), \tag{10}$$

$$\boldsymbol{atte}^1_{i,j,k,n} = \text{softmax}\left(\boldsymbol{z}^1_{i,j,k,n}\right), \tag{11}$$

$$\boldsymbol{x}'_{i,j,k} = \sum_{n=1}^{N_{i,j}} \boldsymbol{atte}^1_{i,j,k,n}\widetilde{\boldsymbol{h}}^a_{i,j,n}, \tag{12}$$

where $\boldsymbol{w}^1_h \in \mathbb{R}^{d\times(d+d_e)}$, $\boldsymbol{w}^1_x \in \mathbb{R}^{d\times d}$, $\boldsymbol{u}^{qo}_{i,k}, \boldsymbol{W}^1_z, b^1 \in \mathbb{R}^d$. The input of the second hop is $\boldsymbol{F}^1_{i,j,k}$.

$$\boldsymbol{F}^1_{i,j,k} = \tanh\left(\boldsymbol{w}_{f_1}\boldsymbol{x}'_{i,j,k} + b_f\right) + \boldsymbol{w}_{f_2}\boldsymbol{F}^0_{i,j,k}, \tag{13}$$

where $\boldsymbol{w}_{f_1} \in \mathbb{R}^{d\times(d+d_e)}$, $\boldsymbol{w}_{f_2} \in \mathbb{R}^{d\times d}$, $b_f \in \mathbb{R}^d$. After iterating T times, we get the Tth fusion vector $\boldsymbol{F}^T_{i,j,k}$. Labels on the ith question's options are predicted simultaneously. The predicted probability on the kth option is calculated below:

$$\boldsymbol{l}'_{i,j,k} = \text{softmax}\left(\boldsymbol{w}\boldsymbol{F}^T_{i,j,k} + b\right), \tag{14}$$

where $\boldsymbol{w} \in \mathbb{R}^{c\times d}$, $b \in \mathbb{R}^c$, c is the number of label categories (c = 3), and $\boldsymbol{l}'_{i,j,k}$ is the predicted probability of the kth option. The UCINet can be trained with the cross entropy loss function:

$$\mathcal{L}_{CE} = -\frac{1}{N_{i,j,k}}\sum_{i,j,k} \boldsymbol{l}_{i,j,k}\log(\boldsymbol{l}'_{i,j,k}). \tag{15}$$

4.1.3. Metric triplet loss

In reverse QA, given a question, there can be various answers with the same choices of all options. We hope that the representations of answers in the same semantics are closer to those in different semantics under the same question. The number of answers having the same choices for a given question is three to seven in the two data sets allowing us to leverage metric learning to aid texts' representations. Thus, we add a metric triplet loss (TL) [34] to UCINet, where a sample comprises a question and answer pair.

The first step is to construct training triplets. Given an anchor x_a comprising a question and one of its answers, a positive sample x_p is from other answers in the question whose label sets are the same as the anchor's, and for a negative sample x_n, its answer from those whose label sets differ from the anchor's. We only choose different semantic answers under the same question without considering the negative samples under different questions. Thus, metric learning can play a more effective role. The triplet loss $\mathcal{L}_{Tri}$ is

$$\mathcal{L}_{Tri} = \sum_i \sum_{(x_a,x_p,x_n)} \max\left(\left\|\boldsymbol{h}^i_{x_a} - \boldsymbol{h}^i_{x_p}\right\| - \left\|\boldsymbol{h}^i_{x_a} - \boldsymbol{h}^i_{x_n}\right\| + m, 0\right), \tag{16}$$

where m is the margin which is a hyper-parameter. $\boldsymbol{h}^i_{x_a}$, $\boldsymbol{h}^i_{x_p}$ and $\boldsymbol{h}^i_{x_n}$ are the representations of the anchor, the positive sample, and the negative sample in the ith question obtained from UCINet which is calculated as follows:

$$\boldsymbol{h}^i_{x_a} = \sum_{k=1}^{|o_i|} F^T_{i,x_a,k}/|o_i|, \tag{17}$$

where $F^T_{i,x_a,k}$ is the Tth fusion vector of the kth option in the ith question about the anchor. $\boldsymbol{h}^i_{x_p}$ and $\boldsymbol{h}^i_{x_n}$ are calculated in the same way. The total loss of UCINet is expressed as follows:

$$\mathcal{L}_{Final} = \mathcal{L}_{CE} + \lambda \times \mathcal{L}_{Tri}, \tag{18}$$

where λ is the weight of triplet loss. We believe that the classification loss and the metric learning loss have caused different

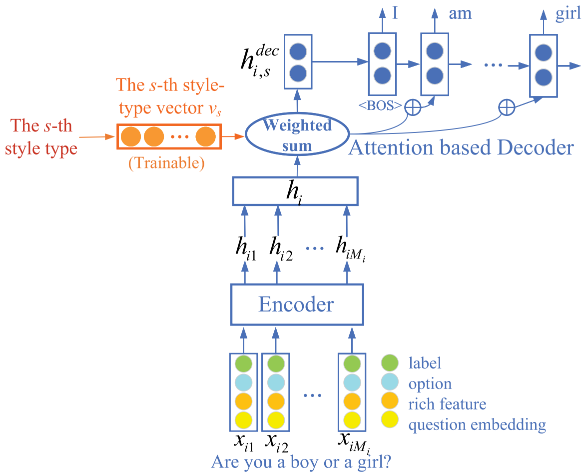

Fig. 3. Structure of SAGNet. When using different style-type vectors v_s, SAGNet can generate answers with diverse styles. Therefore, SAGNet can generate multiple different answers to a question, which is better than existing methods.

focuses of our model. Treating them equally in the training process is not the optimal solution. The model is expected to learn the appropriate representation first and then make the correct classification. The weight is set to $1-\frac{l}{L}$ [35] in our experiments, l is the current training epoch, and L is the total number of epochs. Therefore, in the first few epochs, the model is more concerned about obtaining good text representations, and in the subsequent epochs, the model pays more attention to the classification performance.

4.2. SAGNet

In QA, data augmentation through QG can improve the QA model's performance [4,36]. Inspired by this, we investigate generating diverse answers automatically to augment the training data set of the reverse QA models (i.e., UCINet). Existing methods can only generate a unique answer when a question is given. Thus, the number of generated answers is small, and their quality is not good. In this paper, we propose a style-based answer generation network (SAGNet) wherein texts are determined by both semantics and styles. Even under the same semantics, as long as the input style-type vectors are different, the generated answers will be diverse, which is consistent with the expression diversity of natural language. For example, if the question is "Are you a boy or a girl?", a female user may have answers with different styles such as "I am a girl". and "I am not a boy".

Attention-based seq2seq (A-seq2seq) model and Transformer are two effective solutions for seq2seq problems. Fig. 3 shows SAGNet on the basis of the attention-based seq2seq framework [3]. A style-type vector is added to the intermediate vector between the encoder and the decoder, and the complete vector is fed into the decoder to generate answers. When the encoder and decoder of Transformer replace the encoder and decoder shown in Fig. 3, the network is called SAGNet$_T$, which is explored in our experiments. Taking SAGNet on the basis of A-seq2seq as an example, we introduce the main techniques of the style-based answer generation network.

4.2.1. Style types generation

This network relies on the pre-determined style type for each answer based on the following assumption: ***For each question, answers with the same choices of all options have the same semantics, and they differ in text styles.*** Accordingly, the mean

---

Algorithm 1 Generate each answer's style type.

Input: $\boldsymbol{q}_i$ and $\boldsymbol{a}_{i,j}$, $i=1,2,\cdots,I; j=1,2,\cdots,J_i$.
Parameter: Parameter(s) in the clustering algorithm.
Output: Each answer's style type in the training set.

1: Obtain the feature vector $\boldsymbol{h}_{i,j}$ of each sample from the answer understanding network (UCINet).
2: Calculate the mean value $\overline{\boldsymbol{h}_i^c}$ for each answer group.
3: Calculate the style representation of each sample $\Delta\boldsymbol{h}_{i,j}=\boldsymbol{h}_{i,j}-\overline{\boldsymbol{h}_i^c}$.
4: Cluster the style representations $\Delta\boldsymbol{h}_{i,j}$, and the class index of each sample is its style type.

---

vector of representations of answers in an answer group[2] can be viewed as a semantic vector. The style representation for answer $a_{i,j}$ is obtained by using its feature vector minus its corresponding semantic vector, which is calculated as follows:

$$\Delta\boldsymbol{h}_{i,j}=\boldsymbol{h}_{i,j}-\overline{\boldsymbol{h}_i^c}, \tag{19}$$

where $\Delta\boldsymbol{h}_{i,j}$ is the obtained style representation, $\boldsymbol{h}_{i,j}(=\sum_{k=1}^{|o_i|}F_{ijk}^T/|o_i|)$ is the representation for $a_{i,j}$, $|o_i|$ is the number of options in the ith question, and $\overline{\boldsymbol{h}_i^c}$ is the mean vector of the cth answer group in the ith question.

We calculate all answers' style representations and cluster them to determine the style type of each answer where the number of style types equals the number of clusters. Two clustering methods are adopted in our experiments, namely CFDP [37] and k-means [38]. K-means is a classic clustering method with simple principles and easy implementation. CFDP first proposed on Science improves K-means and DBSCAN [39], which is effective and allows the finding of nonspherical clusters. Other clustering algorithms can also be used according to data characteristics. Style types generation algorithm is shown in Algorithm 1. Each style type corresponds to a trainable style-type vector v_s in the model shown in Fig. 3, which is initialized by a uniform distribution. v_s is the representation vector of the sth style type. Once the net is trained, a set of style-type vectors is obtained.

4.2.2. Encoder of SAGNet

To make the semantics of the generated answers conform to the given labels, the model should encode both the option and label information in the question. The input of SAGNet's encoder includes the question's word embedding ($\boldsymbol{q}_{i,m}$), the label information ($\boldsymbol{l\_r}_{i,m}$), the option indicator ($\boldsymbol{l\_o}_{i,m}$), and the rich feature information ($A_{i,m}^q$).

$$\boldsymbol{x}_{i,m}=\left[\boldsymbol{q}_{i,m},\boldsymbol{l\_r}_{i,m},\boldsymbol{l\_o}_{i,m},\boldsymbol{A}_{i,m}^q\right], \tag{20}$$

where $\boldsymbol{x}_{i,m}$ is the input vector of the encoder. For $\boldsymbol{l\_r}_{i,m}$, 0, 1 and 2 represent that the current word's label is "un-chosen", "chosen" and "uncertain", respectively, and -1 means that the current word in the question has no specified label. The output of the encoder for the mth word is denoted as $\boldsymbol{h}_{i,m}^{enc}$.

$$\boldsymbol{h}_{i,m}^{enc}=\mathrm{BiLSTM}\left(\boldsymbol{h}_{i,m-1}^{enc},\boldsymbol{h}_{i,m+1}^{enc},\boldsymbol{x}_{i,m}\right). \tag{21}$$

4.2.3. Decoder of SAGNet

The input vector of the decoder ($\boldsymbol{h}_{i,s}^{dec}$) is the weighted sum of the last moment vector output by the encoder ($\boldsymbol{h}_{i,M_i}^{enc}$) and the

[2] Answer group is composed of answers in the same semantics.

Table 2
Statistics of Mdata and Tdata. "Average" means the average number of answers corresponding to a question.

Data set	Questions	Answers	Average
Mdata	1006	22383	22.25
Tdata	534	10818	20.26

style-type vector ($\boldsymbol{v}_s$). Let $\boldsymbol{g}_{i,0}$ represent the initial state of the decoder.

$$\boldsymbol{h}_{i,s}^{dec} = \boldsymbol{h}_{i,M_i}^{enc} + \boldsymbol{w}_s \boldsymbol{v}_s + b_s, \tag{22}$$

$$\boldsymbol{g}_{i,0} = \tanh\left(\boldsymbol{w}_d\left(\boldsymbol{h}_{i,s}^{dec}\right) + b_d\right), \tag{23}$$

$$\boldsymbol{g}_{i,t} = \mathrm{BiLSTM}\left(\boldsymbol{y}_{i,t-1}, \boldsymbol{g}_{i,t-1}, \boldsymbol{c}_{i,t}\right), \tag{24}$$

where $\boldsymbol{w}_s, \boldsymbol{w}_d \in \mathbb{R}^{d' \times d'}$, $b_s, b_d \in \mathbb{R}^{d'}$, d' is the dimension of the hidden layers in both encoder and decoder, $\boldsymbol{y}_{i,t-1}$ is the word embedding at the previous moment, and $\boldsymbol{c}_{i,t}$ is a vector obtained by the attention mechanism which is calculated as follows:

$$\boldsymbol{e}_{i,t,m} = \boldsymbol{V}_a^T \tanh\left(\boldsymbol{W}_a \boldsymbol{g}_{i,t-1} + \boldsymbol{U}_a \boldsymbol{h}_{i,m}^{enc}\right), \tag{25}$$

$$\boldsymbol{\alpha}'_{i,t,m} = \mathrm{softmax}\left(\boldsymbol{e}_{i,t,m}\right), \tag{26}$$

$$\boldsymbol{c}_{i,t} = \sum_{m=1}^{M_i} \boldsymbol{\alpha}'_{i,t,m} \boldsymbol{h}_{i,m}^{enc}, \tag{27}$$

where $\boldsymbol{W}_a, \boldsymbol{U}_a \in \mathbb{R}^{d' \times d'}$, and $\boldsymbol{V}_a \in \mathbb{R}^{d'}$. The output of the tth position y_t is predicted as follows:

$$p\left(y_{i,t} \mid y_{i,1}, y_{i,2} \ldots y_{i,t-1}, \boldsymbol{x}_i\right) = \mathrm{softmax}\left(\boldsymbol{W} \boldsymbol{g}_{i,t}\right), \tag{28}$$

where $\boldsymbol{W} \in \mathbb{R}^{vs \times d'}$, and vs is the vocab size. Given the training data (i.e., q_i, $a_{i,j}$, o_i, $l_{i,j}$, and the style type of $a_{i,j}$), the learning goal of SAGNet is to maximize the likelihood of the generation of $a_{i,j}$ based on Eq. (28). Once SAGNet is trained, answers with multiple styles can be generated.

4.2.4. Weighted loss of SAGNet

If the number of answers in each style category is extremely imbalanced in the training set, tail categories may not be learned well. A weighted loss can be adopted to increase the weight of samples in tail categories [40], which can improve the model's performance on tail categories. The weighted loss of SAGNet is defined as follows:

$$\mathcal{L}_w = -\frac{1}{N} \sum_{\mathrm{i}=1}^{N} w_{i_{cc}} \sum_{\mathrm{t}=1}^{T} l_{i,t}, \tag{29}$$

where N is the number of samples in the training data. cc, $w_{i_{cc}}$, and $l_{i,t}$ are the ith sample' category, weight and cross entropy loss of the tth time. The weighting function of the Class-balanced method [40] is used in our experiments.

5. Experiments

5.1. Data sets

Reverse QA is an emerging research problem, and it is different from previous tasks as mentioned in Section 3.2. Except for the two reverse QA data sets mentioned in [1],[3] there are no other data sets for reverse QA. Thus, these two existing data sets, namely, Mdata and Tdata [1] are used in our experiments which are obtained from real applications.

Mdata: This data set contains 1,006 multiple-choice questions. Each question is associated with 22.25 answers on average. The number of options for each question is 2 to 6. In total, there are 22,383 samples in this data set.

Tdata: This data set contains 534 true/false questions. Each question is associated with 20.26 answers on average. In total, there are 10,818 samples in this data set.

For the true/false questions, the question itself is the only option. To unify the two types of questions, the options of true/false questions are set to "None". Hence, the label of each answer in the Tdata includes 0, 1, and 2 which represents the answer's label for the question is "un-chosen", "chosen", and "uncertain", respectively. For Mdata, an answer will have different labels on different options in the question. Similarly, 0, 1 and 2 indicate that the label on the option is "un-chosen", "chosen", and "uncertain", respectively. All texts are Chinese which are segmented by Jieba library. During the experiments, the training/validation/test set division of AntNet's [1] experiments is followed. The statistics of the data sets are summarized in Table 2.

5.2. Implementation details

In our experiments, all word and char vectors are initialized with 256-d pre-trained word and char embeddings using GloVe [41] unless noted, as previous works did [15,17,42]. All out-of-vocabulary words are initialized by sampling from the uniform distribution. The hidden dimensions of BiLSTM, Transformer, and the attention layer are all set to 256. The sizes of the NER dictionary and the POS dictionary in the rich feature information are 4 and 29. The dimension of the word vector pre-trained by BERT [43] and mT5 [44] are both 768. In addition, the numbers of the block and head in Transformer are set to 6 and 8.

We use grid search to search the batch size, learning rate, and dropout values. In UCINet, the batch size, learning rate, and dropout are set to 16, 5×10^{-4} and 0.2, respectively. We truncate the lengths of questions and answers to 33. The hop value of the multi-hop attention is set to 3, and the number of epochs is set to 20. In SAGNet, the batch size is set to 16. Dropout is applied at a rate of 0.2. The learning rate is 2×10^{-4}. We truncate the lengths of texts to 64. The number of BiLSTM layers in both encoder and decoder is set to 2, and the number of epochs is set to 20.

We minimize the loss function using the ADAM [45] optimizer. To prevent the gradient from exploding during training, the maximum gradient is set to 10. All mentioned models are trained with Tensorflow.

5.3. Evaluation

In this part, experiments are designed to verify the effect of UCINet and SAGNet on the basis of Mdata and Tdata.

5.3.1. Results on users' choices inference

UCINet infers a user's choices of all options in a machine-raised question simultaneously, which can help the machine obtain users' intents accurately.

Evaluation indicators

Accuracy and Micro-F1 are used to evaluate the performance of the models. For a fair comparison with existing methods, the calculation of accuracy is consistent with the existing work, which is as follows:

$$acc = \frac{\sum_{i=1}^{I}(1/J_i)\sum_{j=1}^{J_i}(1/|o_i|)\sum_{k=1}^{|o_i|} I_{i,j,k}}{I}, \tag{30}$$

$$I_{i,j,k} = \begin{cases} 1 & \text{if } l_{i,j,k} = l'_{i,j,k} \\ 0 & \text{else} \end{cases}. \tag{31}$$

Macro-F1 is also used whose calculation follows Takahashi et al. [46].

[3] https://github.com/NlpResearch/AntNet-rverseQA/tree/master/data

Table 3
Comparisons with baseline models on Mdata and Tdata. The best results of GloVe/BERT/mT5-based models are all in bold. w/o means "without". TL means "triplet loss". MHA means "multi-hop attention mechanism". OE means "option encoding module". RV means "reward vector module". (A) means that only answers are encoded into the model. (Q+A) means that both answers and questions are encoded into the model.

Word Embedding	Methods	Acc(Mdata)%	F1(Mdata)%	Acc(Tdata)%	F1(Tdata)%
Glove	BiLSTM (Q+A) [47]	70.05	69.88	72.96	72.78
	BiLSTM (A) [47]	67.01	66.81	73.75	71.96
	Transformer (Q+A) [8]	69.66	65.37	71.67	69.11
	Transformer (A) [8]	67.41	65.25	74.35	73.43
	ATAE [16]	68.65	69.24	74.58	73.61
	RAM [17]	71.21	70.88	75.03	76.35
	IAN [42]	70.86	69.78	74.85	75.21
	Semi-IAN [15]	72.63	73.87	75.25	74.98
	AntNet [1]	82.13	82.86	79.21	78.25
	UCINet with TL	**92.75**	**92.54**	**87.08**	**87.11**
Ablated UCINet	UCINet w/o MHA	89.95	89.78	85.21	85.42
	UCINet w/o OE	90.05	88.65	84.78	83.92
	UCINet w/o RV	91.76	92.55	85.95	84.66
	UCINet w/o TL	90.73	90.23	80.56	80.88
BERT	UCINet w/o TL	91.35	91.65	82.08	81.57
	UCINet with TL	**93.43**	**92.58**	**87.52**	**86.72**
mT5	UCINet w/o TL	91.74	91.16	82.97	82.23
	UCINet with TL	**93.95**	**92.76**	**88.33**	**87.21**

Performance comparison

To achieve the comprehensive and comparative analysis of our network, we compare our model with a series of advanced models which are frequently used in QA, ABSA, reverse QA, and intent detection:

- *BiLSTM* [47] is a combination of the forward and backward LSTM, which is often used as the feature extractor in natural language processing tasks such as intent detection. In our experiments, BiLSTM adopted in the intent detection task [29] is followed.
- *Transformer* [8] completely relies on the attention mechanism and has made significant improvements in NLP tasks such as machine translation. For the intent detection tasks, the encoder of Transformer is used as the feature extractor [30].
- *RAM* [17] leverages the hidden vectors of BiLSTM as memory vectors. GRU is used to construct a multi-hop-based fusion of the memory vectors and the input target vector. The final dense vector contains information from both sentences and targets. In our experiments, we take question texts as target texts.
- *ATAE* [16] is proposed for the ABSA tasks which is on the basis of BiLSTM. The target vector is concatenated with the embedding of each word. In our experiments, the question texts are taken as the target texts.
- *IAN* [42] uses LSTM to model targets and contexts separately and uses the attention mechanism to model the interactive information of them.
- *Semi-IAN* [15] is a slight variation of IAN, and the difference is that Semi-IAN only retains one-side attention on the basis of the task's characteristics. It is the first method related to answer understanding in reverse QA.
- *AntNet* [1] considers the interaction between answers and questions more effectively, which mainly includes three key modules, namely, unsupervised skeleton extraction for questions, relevance-aware answer representation, and multi-hop-based fusion.

In addition, we concern about UCINet's performances with and without triplet loss represented by UCINet with TL and w/o TL, respectively. The results are shown in Table 3, which are the average values across five runs of training with different seeds.

Overall, our proposed UCINet outperforms all other competitors significantly, especially on Mdata. BiLSTM and Transformer used in intent detection tasks [29,30] do not perform well in reverse QA. These methods adopt deep learning models to obtain text representations. The softmax function is applied to the representations of linear transformation to give the probability distribution over the intent labels. In reverse QA, the interaction among questions, answers, and options is crucial, but this information is not taken into account seriously in intent detection models. As mentioned earlier, metric learning is very suitable for our data set. Therefore, the triplet loss improves the performance of our model. Using BERT [43] and mT5 [44] pre-trained models, the performance of UCINet is further enhanced. The superiority of UCINet lies in the following reasons: (1) UCINet predicts the labels on multiple options simultaneously. Therefore, the model can learn the intrinsic relationship among the user's choices of different options. (2) Metric learning helps the model learn better text representations which are the basis for the model to play an effective role. (3) UCINet models the interactive information of questions and answers from multiple aspects, including the multi-hop fusion mechanism and the relevance reward score of the answer. In addition, since UCINet predicts the labels on multiple options simultaneously, the repeated input of questions and answers is reduced. Thus, the prediction efficiency of our model increased by 41.5% compared with AntNet.

The use of triplet loss is especially helpful. The accuracy and loss of UCINet with and w/o TL on Mdata during the training phase are shown in Fig. 4. UCINet with TL has smaller loss and higher accuracy, and its loss converges faster, indicating the superiority of our method.

The ablation study of UCINet

To investigate the impact of some critical components such as the option encoding module (OE), the multi-hop attention mechanism (MHA), the question–answer correlation reward vector module (RV), and the triplet loss (TL), we perform the comparison between the full UCINet model and its ablations in Table 3. Note that, w/o MHA means that the model uses the 1-hop attention mechanism instead of the multi-hop attention mechanism.

From the results, the model with MHA gains a significant improvement compared to the model w/o MHA, which shows that the multi-hop attention mechanism has a solid ability to fuse information. Comparing the results of UCINet with and w/o

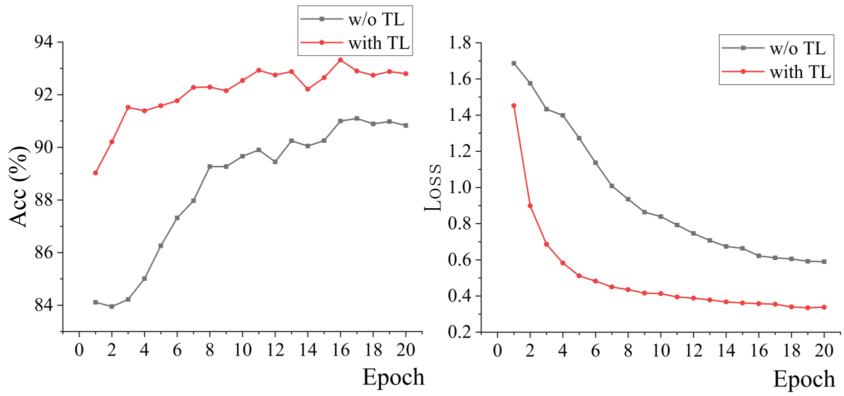

Fig. 4. The accuracy (left) and loss (right) curves of UCINet with and w/o TL during the training phase. UCINet with TL has higher accuracy, and it converges faster than the model w/o TL.

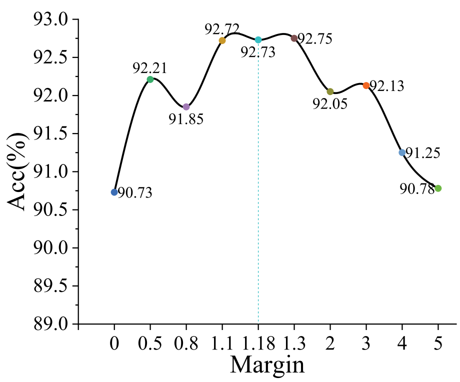

Fig. 5. Accuracies under different margin values.

TL, we observe that the performance of UCINet w/o TL is incomparable with UCINet with TL. It indicates the importance of metric learning, making the model learn better representations. Besides, UCINet w/o OE performs worse than UCINet w/o RV, which verifies that the OE module is more effective than the RV module. Both OE and RV modules improve the performance of UCINet. In summary, the results show that all four discarded components are crucial for good performance.

Analysis of the margin value in metric learning

An appropriate margin value can enhance the performance of metric learning. Specifically, the larger the margin value, the stricter the requirement of metric learning. However, when the margin is too large, the inequality $\left\|\boldsymbol{h}_{x_a}-\boldsymbol{h}_{x_p}\right\|+m \leq \left\|\boldsymbol{h}_{x_a}-\boldsymbol{h}_{x_n}\right\|$ may not be held for most samples. If this inequality is not held for all triplets, the metric triplet loss will become invalid. We observe that the average distance between any two representations in Mdata is 1.18. Thus, candidate margin values close to 1.18 are also evaluated. The models' performances under different margins are compared, as shown in Fig. 5. The model achieves the best performance when the margin equals 1.3. When it is close to the average distance, the performances of these values are better than those of other margin values. Therefore, when using triplet loss, values close to the average distance among all samples in the data set are good choices for margins. When the margin equals 3, 35% triplets satisfy the inequality. If it increases, the number of triplets satisfying the inequality will decrease. Thus, the effect of the triplet loss will drop until it is valid. Similar observations are also obtained on Tdata. The conclusion is that when triplets are satisfying the requirement of the triplet loss, the triplet loss will have a positive impact on our model. Otherwise, it is invalid.

Table 4
The comparison of BLEU and ROUGE values of answers generated by A-seq2seq and SAGNet. For A-seq2seq, only one answer can be generated for a question. For SAGNet, multiple answers can be generated for a question. TopK means selecting k answers with the best quality (answers with the highest BLUE and ROUGE values). RandomK means selecting k answers randomly.

Methods		BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-L
Attention-based seq2seq		0.63	0.45	0.30	0.19	0.33	0.33
SAGNet (Ours)	Top1	0.84	0.73	0.59	0.46	0.51	0.50
	Top3	0.80	0.66	0.50	0.37	0.46	0.45
	Top5	0.77	0.62	0.44	0.32	0.41	0.40
	Top7	0.75	0.57	0.39	0.27	0.37	0.38
	Top9	0.63	0.46	0.31	0.18	0.31	0.32
	Random5	0.64	0.45	0.32	0.19	0.34	0.35

5.3.2. Results on answer generation

SAGNet aims to generate answers with multiple styles based on different style-type vectors. The generated answers are used to augment the training data set for UCINet's training.

Quality evaluation for generated answers

Automatic Machine Evaluation: BLEU and ROUGE are two commonly used evaluation indicators in natural language generation, which are introduced as follows:

- BLEU [48] measures precision by calculating how many words in predictions appear in references. BLEU-1 to BLEU-4 use 1-gram to 4-gram for calculation, respectively.
- Rouge-1 and Rouge-L [49] measure recall by calculating how many words in reference sentences appear in predictions using 1-gram and Longest Common Sub-sequence (LCS).

For each sample, existing methods, including the attention-based seq2seq [3] can only generate one answer for a question, SAGNet can generate multiple answers in various styles with different style-type vectors. Thus, more answers of high quality can be generated. The number of generated answers for each sample equals the number of style-type vectors. CFDP clustering method is used here. When the average number of neighbors is around 1%–2% of a total number of points in the data set, CFDP achieves a good performance [37]. Based on this empirical principle, the hyper-parameter d_c in CFDP is set to 0.25/0.15 for Mdata/Tdata with the help of Dunn [50] index. After that, the style types for Mdata and Tdata are clustered into 10 and 12 categories, respectively. TopK answers with the best quality (answers with the highest BLEU and ROUGE values) are selected in each sample's all generated answers. The comparison results on Mdata are shown in Table 4.

The BLEU and ROUGE values achieved by SAGNet are higher than those of the baseline before Top9, indicating that our model

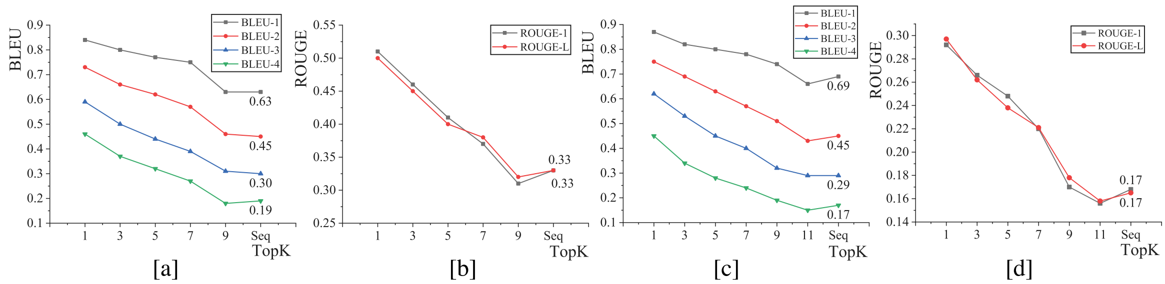

Fig. 6. The quality comparison of filtering different number of generated answers on Mdata ([a] and [b]) and Tdata ([c] and [d]). "Seq" refers to the attention-based seq2seq model.

Table 5
Results of human evaluation. The results of clustering serve as the style labels for human answers.

	Fluency	Complexity	Style matching	Semantic preservation
Human	4.41 ± 0.12	2.89 ± 0.15	4.24 ± 0.16	/
SAGNet	4.01 ± 0.18	2.94 ± 0.16	3.98 ± 0.18	4.08 ± 0.17
A-seq2seq	3.43 ± 0.19	2.58 ± 0.18	/	3.15 ± 0.24

Table 6
The comparison of BLEU and ROUGE values of answers generated by the original Transformer and SAGNet$_T$. Only one answer can be generated for a question for the original Transformer. After incorporating the style-type vectors, various answers can be generated for a question.

Methods		BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-L
Transformer		0.65	0.46	0.31	0.19	0.33	0.34
SAGNet$_T$	Top1	0.85	0.74	0.60	0.47	0.51	0.50
	Top3	0.81	0.67	0.50	0.39	0.45	0.46
	Top5	0.76	0.63	0.43	0.32	0.41	0.42
	Top7	0.73	0.58	0.40	0.27	0.37	0.38
	Top9	0.66	0.45	0.30	0.20	0.32	0.33
	Random5	0.67	0.46	0.32	0.21	0.34	0.35

is better than A-seq2seq in precision and recall. When only the Top1 answers generated by SAGNet are selected, the BLEU-4 and ROUGE-L values can reach 0.46 and 0.50 which have increased by 0.27 and 0.17 compared to the values of A-seq2seq. Table 4 presents that until we select the best answers to Top7, the answers generated by our model are still better than those generated by the baseline, that is, in the case of better quality, the ratio of the number of answers generated by SAGNet and A-seq2seq can reach at least 7:1. Therefore, SAGNet can generate more answers of good quality. The results of randomly selecting five answers are also shown in the table. Since the large number and diversity of generated answers, we recommend using answers of better quality. Also, the performance comparison curves are shown in Fig. 6, which indicates that the quality of answers generated by SAGNet remains much better than those generated by A-seq2seq on true/false questions.

Human Evaluation: In the text generation tasks, it is difficult to automatically evaluate the quality of the generated texts [51]. We adopt a method that combines automatic machine evaluation and human evaluation.

For human evaluation, we evaluate the quality of the generated answers from four perspectives: fluency, complexity, style matching degree, and semantic preservation degree. We randomly select 400 question–answer pairs from the data set and the answers generated by SAGNet and A-seq2seq. For answers generated by SAGNet, we choose the Top3 answers. Four Chinese professionals are asked to rate the answers (1–5 points scale) from four perspectives above. The evaluation criteria for the four metrics are as follows:

- Fluency: Could this sentence be spoken by a native Chinese speaker?
- Complexity: Is the vocabulary and structure of this sentence complicated?
- Style matching degree: How well does the answer's style match the given style type?
- Semantic preservation degree: How well does the semantics of the answer match the given label set?

Table 5 shows the average scores of answers generated by the human and machine. The quality of answers generated by SAGNet slightly surpasses that of human answers in terms of complexity but is inferior to human answers in fluency and style matching. It exceeds that of answers generated by A-seq2seq regarding all the metrics. Although there are still unsatisfactory answers, our model can generate answers having good quality and conforming to the given labels. SAGNet has absolute advantages over the baseline model.

Analysis of SAGNet$_T$

Transformer is another effective solution to the seq2seq problem. Thus, we apply the idea of the style-based answer generation to Transformer. The encoder and decoder shown in Fig. 3 are replaced by the encoder and decoder of Transformer. The value of d_c is the same as before. The quality of answers generated by the original Transformer and SAGNet$_T$ are compared in Table 6. The BLEU and ROUGE values achieved by SAGNet$_T$ are higher than those of the original Transformer before Top9. It inspires us that using style-type vectors to generate multiple answers with diverse styles, and selecting high-quality answers from the generated ones is an effective way to obtain more answers in high quality.

Analysis of the number of style types

In SAGNet, the number of style-type vectors equals that of style clusters. Using K-means, the number of style clusters is set to 10, 20, 30, 40, and 50 when generating answers. According to the observations of generated answers, we find that if the number of categories is too large, the boundaries between any two categories may be blurred, and if it is too small, no obvious characteristics can be observed in each category. ROUGE and BLEU values of Top3 answers in each case on Mdata are compared in Fig. 7. When the number of styles is 10, the quality of the generated answers is better which is consistent with the results of CFDP. Therefore, dividing the text styles into ten categories for Mdata can generate answers with high BLEU and ROUGE values.

Analysis of style categories

After clustering the answers' style representations in Mdata into ten categories, we count the category proportions of answers in Mdata, shown in Fig. 8[a]. Answers in category 0 account for

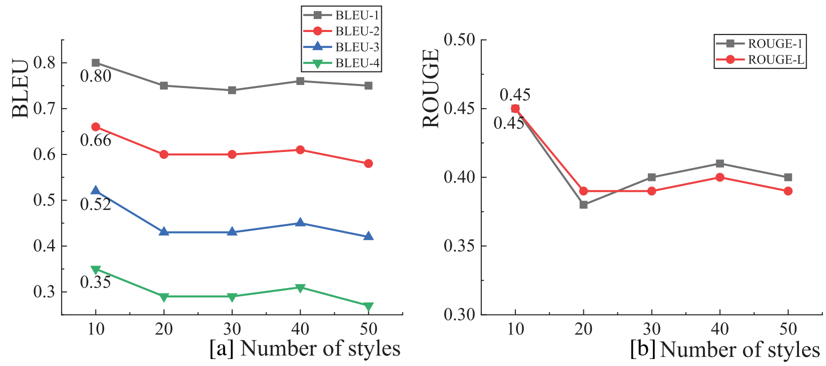

Fig. 7. BLEU ([a]) and ROUGE ([b]) values of generated answers with different number of style types on Mdata. When the number of style clusters equals ten, the model performs best.

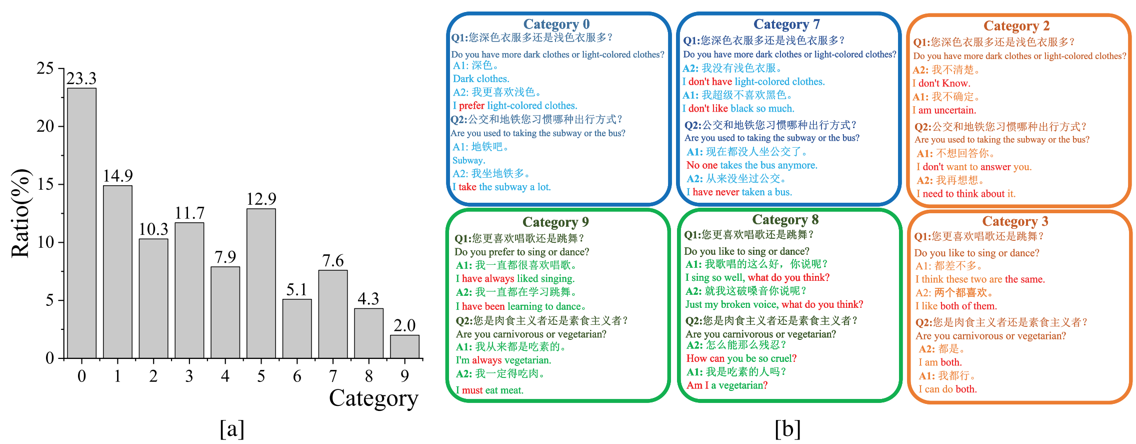

Fig. 8. [a] represents the proportion of answers in each category in Mdata. K-means is used here. [b] shows examples of answers in six categories, and the common features of answers in each category are marked in red. Boxes with the same colors (green and blue) show answers with the same semantics but different styles. The two orange boxes show two categories of answers with special styles. Answers in one category do not give clear attitudes toward all the options, and answers in the other category have positive attitudes toward all options.

the most where the answers are generally concise and clear. We analyze the generated answers in different categories and find that the generated answers are in different styles when using different style-type vectors. Thus, it verifies our assumption is correct. Fig. 8[b] shows some generated answers in six categories, showing that each category has common characteristics.

We fully summarize and analyze the answers in each category and get the following findings. Answers in Category 0 are straightforward and clear. Answers in Category 1 states the advantages of the selected option. Answers in Category 2 do not give preferences that are similar to “I don’t know”. and “I don’t want to choose”., etc. Answers in Category 3 have positive attitudes toward all options. In addition, the language style of Category 4 is relatively relaxed where the answers frequently contain interjections. Answers in Category 5 are more obscure and are mostly explanatory sentences. Contrary to those in Category 3, answers in Category 6 have negative attitudes toward all options. Most of the answers in Category 7 deny the option which is not chosen. They often contain negative words like “no”, “don’t”, etc. Answers in Category 8 are mostly in the form of rhetorical questions. The tone of answers in Category 9 is very certain, generally containing words such as “always”, “forever”, etc.

Analysis of the weighted loss

Fig. 8[a] indicates that the number of answers in Mdata for each style category is imbalanced. To decrease the gap, a weighted loss can be adopted by SAGNet. The weighting function of the

Table 7
Generated answers with the same semantics and different styles.

Q: 您更喜欢文学类书籍还是艺术类书籍? Do you prefer **literary books** or **art books**?	
A1: 我比较喜欢画画。 I like drawing.	Category 5
A2: 我喜欢艺术类的书籍。 I like art books.	Category 0
A3: 文学类书籍很难懂。 Literary books are difficult to understand.	Category 7
A4: 当然是看艺术类的书籍啦。 Of course, I choose to read art books.	Category 4
A5: 艺术类书籍更有趣。 Art books are more interesting.	Category 1

Class-balance method [40] is used. The BLUE-4 and ROUGE-L values of the generated answers in the five tail categories of the models with and w/o weighted loss are compared in Fig. 9.

After using the weighted loss, the quality of the generated answers in tail categories is improved, especially the last tail category. The reason is that assigning higher weights to samples in tail categories increases the contribution of tail categories to the model. Therefore, using weighted loss is an effective way to improve the model’s performance on imbalanced data.

Analysis to the generated answers

This experiment shows that SAGNet can generate answers with the same semantics and different styles. The text styles do not have a clear definition, but we can summarize the common characteristics of answers in each category. For example, the text

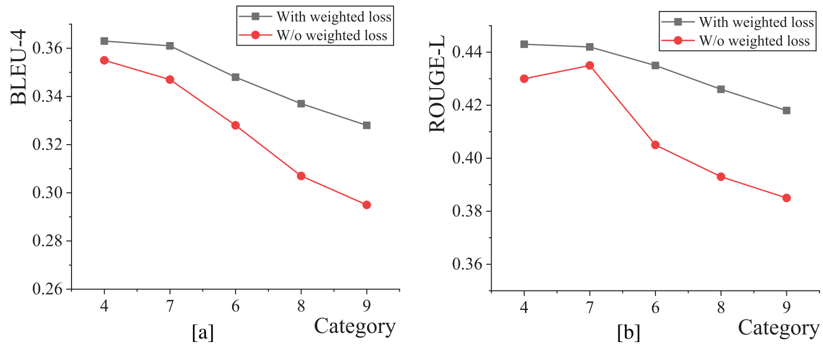

Fig. 9. The effect of the weighted loss on Mdata. Top3 answers are selected. The horizontal axis is sorted according to the proportion of each category shown in Fig. 8[a].

Table 8
Examples of more generated answers.

Q1: 假期您准备回家还是出去旅游?? Are you going to go home or travel during the holidays? **Option set:** {go home,go travel}			**Q2:** 工作和家庭哪个对您来说更重要?? Which is more important to you, work or family? **Option set:** {work, family}		
Answer	**Label set**	**Category**	**Answer**	**Label set**	**Category**
A1: 旅游人太多了。 There are too many tourists.	{1,0}	7	**A1:** 当然是家庭啊。 Family, of course.	{0,1}	4
A2: 肯定出去旅游。 I will go travel.	{0,1}	0	**A2:** 一切都是为了家人。 Everything is for the family.	{0,1}	5
A3: 好久都没回家了。 I have not been home for a long time.	{1,0}	5	**A3:** 只有工作才能让我快乐。 Only work can make me happy.	{1,0}	1
A4: 我不确定。 I am not sure.	{2,2}	2	A4: 都重要。 Both of them are important.	{1,1}	3

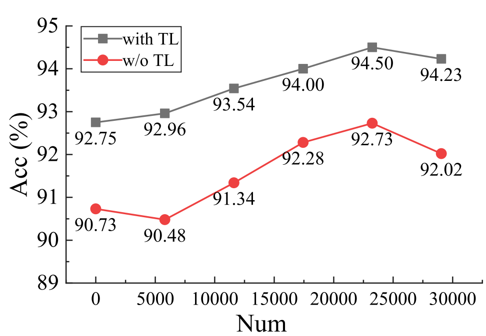

Fig. 10. Accuracies of UCINet with and w/o TL under adding different number of generated answers to Mdata.

style can refer to whether the expression is implicit or direct, whether the emotional attitude is certain or speculative, etc. Taking the sample "**question: Do you prefer literary books or art books?; option set:{"literary books", "art books"}; label set:{0,1}**", in the case of the same semantics (the option of "art books" is chosen), our model can generate answers with multiple styles. Examples of generated answers are shown in Table 7. The options described in these answers are different. Some are literary books, while others are art books. The expression styles and emotional attitudes of these answers are also different. However, the semantics of these answers are the same.

From Table 7, Answer 1 expresses the user's preference but does not directly make a choice. Answers 2 and 4 are direct and Answer 4 is more relaxed. Answers 3 states the shortcoming of the option which is not chosen (literary books). Answer 5 talks about the advantage of the chosen option (art books). Table 7 shows that SAGNet can generate answers with different expression styles even under the same semantics. Table 8 shows more answers generated by SAGNet, which indicates that SAGNet can generate sentences with various styles and rich content.

Analysis to the effect of augmenting training data set

The cost of obtaining human-labeled data is high. Thus, we augment the data set by automatically generating answers with different styles. A different number of answers in good quality generated by SAGNet are selected to augment the training set. The results of UCINet with and w/o TL after augmenting the Mdata are shown in Fig. 10.

Answers generated by SAGNet to amplify the data set improve the performance of both UCINet with and w/o TL. The highest increases for UCINet with TL and w/o TL are 1.75% and 2%. The original data set contains 22,383 samples in Mdata. We find that when 23,240 generated answers are added (the ratio of human answers to generated answers is close to 1:1), the model performs best. We also verify that the generated answers are effective on Tdata.

6. Conclusion

Reverse QA is an essential part of the human–machine dialogue which is widely used in real life. To increase the naturalness of human–machine dialogues, this paper studies reverse QA from two perspectives, namely, automatically inferring the users' choices of options and the style-based answer generation. We propose UCINet which infers users' choices of options in machine-raised questions accurately and efficiently according to the users' answers. Metric learning is leveraged to produce better text representations which further improves the inference performance. To automatically generate answers, SAGNet is designed wherein texts are considered to be determined by both semantics and styles. Therefore, SAGNet can generate multiple answers with various styles for a question that is better than existing methods. The generated answers can subsequently augment the training set. Experimental results show that UCINet outperforms the existing methods significantly. SAGNet can generate more answers with different styles in better quality, and the generated answers are verified to be beneficial for UCINet's training. The style-based text generation algorithm can be used in other text generation models and is easy to be implemented. Our work can be widely used in various human–machine dialogue systems and make them work in a more friendly and natural way.

Declaration of competing interest

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

Acknowledgments

This study is supported by ZJFund 2019KB0AB03, NSFC 62076178, TJ-NSF (19JCZDJC31300, 19ZXAZNGX00050).

References

[1] R. Yao, L. Hou, L. Yang, J. Gui, Q. Yin, O. Wu, Deep human answer understanding for natural reverse QA, 2019, arXiv preprint arXiv:1912.00398.
[2] C. Liu, S. He, K. Liu, J. Zhao, Curriculum learning for natural answer generation, in: Proceedings of the T Wenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, 2018, pp. 4223–4229.
[3] X. Du, J. Shao, C. Cardie, Learning to ask: Neural question generation for reading comprehension, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1342–1352.
[4] M.A. Sultan, S. Chandel, R.F. Astudillo, V. Castelli, On the importance of diversity in question generation for QA, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5651–5656.
[5] M. Tan, C. dos Santos, B. Xiang, B. Zhou, LSTM-based deep learning models for non-factoid answer selection, 2015.
[6] A.W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, Q.V. Le, QANet: Combining local convolution with global self-attention for reading comprehension, 2018.
[7] Y. Zhang, W. Lu, W. Ou, G. Zhang, X. Zhang, J. Cheng, W. Zhang, Chinese medical question answer selection via hybrid models based on CNN and GRU, Multimedia Tools Appl. 79 (2020) 14751–14776.
[8] V. Ashish, S. Noam, P. Niki, U. Jakob, J. Llion, G.A. N., K. undefinedukasz, P. Illia, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.
[9] W. Zhang, Y. Li, S. Wang, Learning document representation via topic-enhanced LSTM model, Knowl.-Based Syst. 174 (2019) 194–204.
[10] T. Hao, W. Xie, Q. Wu, H. Weng, Y. Qu, Leveraging question target word features through semantic relation expansion for answer type classification, Knowl.-Based Syst. 133 (2017) 43–52.
[11] Y. Tay, L.A. Tuan, S.C. Hui, Multi-cast attention networks, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2299–2308.
[12] J. Rao, L. Liu, Y. Tay, W. Yang, P. Shi, J. Lin, Bridging the gap between relevance matching and semantic matching for short text similarity modeling, in: 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, 2019, pp. 5370–5381.
[13] C. Miao, Z. Cao, Y. Tam, Keyword-attentive deep semantic matching, 2020, arXiv preprint arXiv:2003.11516.
[14] O. Chergui, A. Begdouri, D. Groux-Leclet, Integrating a Bayesian semantic similarity approach into CBR for knowledge reuse in community question answering, Knowl.-Based Syst. 185 (2019) 104919.
[15] Q. Yin, G. Luo, X. Zhu, Q. Hu, O. Wu, Semi-interactive attention network for answer understanding in reverse-QA, in: Proceedings of the 23rd Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2019, pp. 3–15.
[16] Y. Wang, M. Huang, L. Zhao, X. Zhu, Attention-based LSTM for aspect-level sentiment classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 606–615.
[17] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent attention network on memory for aspect sentiment analysis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 452–461.
[18] S. Hu, L. Zou, Z. Zhu, How question generation can help question answering over knowledge base, in: Natural Language Processing and Chinese Computing - 8th CCF International Conference, NLPCC 2019, 2019, pp. 80–92.
[19] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014, pp. 3104–3112.
[20] B.P. Thomas, J. Staiano, Self-attention architectures for answer-agnostic neural question generation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 6027–6032.
[21] W. Chen, P. Li, H.P. Chan, I. King, Dialogue summarization with supporting utterance flow modelling and fact regularization, Knowl.-Based Syst. 229 (2021) 107328.
[22] J. Lu, X. Sun, B. Li, L. Bo, T. Zhang, BEAT: Considering question types for bug question answering via templates, Knowl.-Based Syst. 225 (2021) 107098.
[23] D. Wang, A domain-specific question answering system based on ontology and question templates, in: 2010 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2010, pp. 151–156.
[24] T. Holtgraves, Automatic intention recognition in conversation processing, J. Memory Lang. 58 (3) (2008) 627–645.
[25] T.-E. Lin, H. Xu, A post-processing method for detecting unknown intent of dialogue system via pre-trained deep neural network classifier, Knowl.-Based Syst. 186 (15) (2019).
[26] P. Haffner, G. Tur, J.H. Wright, Optimizing SVMs for complex call classification, in: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003, pp. 632–635.
[27] R. Sarikaya, G.E. Hinton, B. Ramabhadran, Deep belief nets for natural language call-routing, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, 2011, pp. 5680–5683.
[28] G. Tur, D. Hakkani-Tur, L. Heck, What is left to be understood in atis? in: 2010 IEEE Spoken Language Technology Workshop, 2010, pp. 19–24.
[29] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, Y.-N. Chen, Slot-gated modeling for joint slot filling and intent prediction, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 753–757.
[30] J. Wang, K. Wei, M. Radfar, W. Zhang, C. Chung, Encoding syntactic knowledge in transformer encoder for intent detection and slot filling, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 13943–13951.
[31] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[32] H.H. Do, P. Prasad, A. Maag, A. Alsadoon, Deep learning for aspect-based sentiment analysis: A comparative review, Expert Syst. Appl. 118 (15) (2019) 272–299.
[33] M. Yang, W. Tu, Q. Qu, W. Zhou, Q. Liu, J. Zhu, Advanced community question answering by leveraging external knowledge and multi-task learning, Knowl.-Based Syst. 171 (2019) 106–119.
[34] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: A unified embedding for face recognition and clustering, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, 2015, pp. 815–823.
[35] Y. Wang, W. Gan, J. Yang, W. Wu, J. Yan, Dynamic curriculum learning for imbalanced data classification, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 5016–5025.
[36] Y. Sun, D. Tang, N. Duan, T. Qin, S. Liu, Z. Yan, M. Zhou, Y. Lv, W. Yin, X. Feng, B. Qin, T. Liu, Joint learning of question answering and question generation, IEEE Trans. Knowl. Data Eng. 32 (5) (2020) 971–982.
[37] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[38] J.A. Hartigan, M.A. Wong, Algorithm AS 136: A K-means clustering algorithm, J. R. Stat. Soc. 28 (1) (1979) 100–108.
[39] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, Knowl. Discov. Data Min. 96 (34) (1996) 226–231.
[40] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, in: 2019 IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9260–9269.
[41] J. Pennington, R. Socher, C. Manning, Answer-focused and position-aware neural question generation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3930–3939.
[42] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, in: The 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, 2017, pp. 4068–4074.
[43] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[44] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mT5: A massively multilingual pre-trained text-to-text transformer, 2020, arXiv preprint arXiv:2010.11934.
[45] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014.
[46] K. Takahashi, K. Yamamoto, A. Kuchiba, T. Koyama, Confidence interval for micro-averaged F1 and macro-averaged F1 scores, Appl. Intell. (2021) 1–12.

[47] A. Graves, S. Fernandez, J. Schmidhuber, Bidirectional LSTM networks for improved phoneme classification and recognition, in: Proceedings of the 15th International Conference on Artificial Neural Networks, ICANN 2005, 2005, pp. 799–804.
[48] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 311–318.
[49] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, 2004, pp. 74–81.
[50] J.C. Dunn, Well-separated clusters and optimal fuzzy partitions, J. Cybern 4 (1) (1974) 95–104.
[51] A. Gatt, E. Krahmer, Survey of the state of the art in natural language generation: Core tasks, applications and evaluation, J. Artificial Intelligence Res. 61 (2018) 75–170.