AntNet: Deep Answer Understanding Network for Natural Reverse QA

Lei Yang, Qing Yin, Linlin Hou, Jie Gui, Ou Wu*, and James Kwok, *Fellow, IEEE*

***Abstract*—This study refers to a reverse question answering (reverse QA) procedure, in which machines proactively raise questions and humans supply answers. This procedure exists in many real human-machine interaction applications. A crucial problem in human-machine interaction is answer understanding. Existing solutions rely on mandatory option term selection to avoid automatic answer understanding. However, these solutions lead to unnatural human-computer interaction and harm user experience. To this end, this study proposed a novel deep answer understanding network, called AntNet, for reverse QA. The network consists of three new modules, namely, skeleton extraction for questions, relevance-aware representation of answers, and multi-hop based fusion. As answer understanding for reverse QA has not been explored, a new data corpus is compiled in this study. Experimental results indicate that our proposed network is significantly better than existing methods and those modified from classical natural language processing (NLP) deep models. The effectiveness of the three new modules is also verified.**

***Index Terms*—Question Answering (QA), Reverse QA, Answer Understanding, Attention, Long Short-Term Memory (LSTM)**

I. Introduction

AUTOMATIC question answering (QA) is a crucial component in many human-machine interaction systems, such as intelligent customer service, as it can provide a natural way for humans to acquire information [1]. Therefore, QA has received increasing attention in academic research and industry communities in recent years [2]. Questions are solely raised by humans, and answers are then returned by machines in the conventional QA scenario. How to select the best matched answer is the key problem in this setting [3].

Nevertheless, machines are also required to determine human needs or perceive human states in human-machine interaction systems. In such scenarios, machines proactively raise questions, and humans supply answers. This procedure is called reverse QA. Although reverse QA receives little attention in previous literature, it is common in commercial intelligent customer service systems. Fig. 1 shows a reverse QA example from Facebook Job Bot[1]. In almost all these commercial systems, the answer items (e.g., "Find jobs", "Profile", "Job alert", and "Info" in Fig 1) are fixed, and humans are only allowed to select one or more of the fixed candidate items. This strategy is an engineering solution leading that the interaction between user and AI system is not nature.

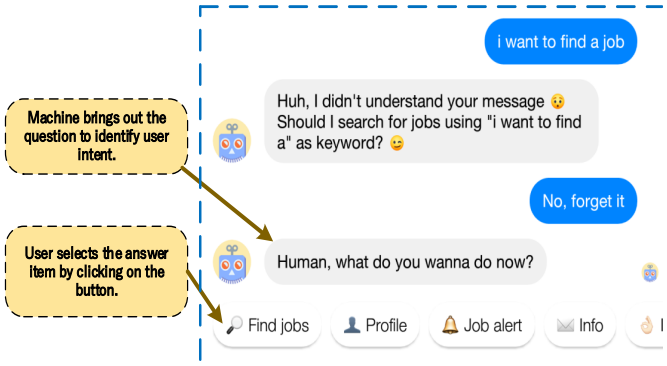

Fig. 1. Reverse QA in a commercial human-AI interaction system. Users cannot type texts for the machine questions. Instead, they are only allowed to select option items (e.g., "Find jobs").

To ensure a nature human-machine interaction and improve user experience, humans should be allowed to type any texts like natural talks in daily life, and machines must automatically understand the meaning of their answers without requiring them to choose fixed options shown in Fig. 1. However, the automatic answer understanding in reverse QA has not been explored so far[2].

In this work, a new deep neural network, called answer understanding network (AntNet), is proposed on the basis of the observations on the new data corpus and inspired by related studies, such as aspect-based sentiment analysis [5], [6].

Given the machine-question and human-answer pair, the AntNet extracts dense feature vectors for the questions and answers, and then fuses these two extracted vectors. Finally, a high-level dense feature vector is obtained and fed into a softmax layer for final answer understanding. Three new modules are included in the AntNet. The first module is the skeleton extraction for questions. The second module is the relevance-aware representation of answers. The primary goal of these two modules is to exclude less important or even disturbing information contained in questions and answers. In addition, the multi-hop fusion module is used to fuse answers and questions vectors. Our proposed network is compared with existing methods and those with a slight modification from classical NLP deep models, such as Transformer [7].

Lei Yang, Qing Yin, Linlin Hou, and Ou Wu are in Center for Applied Mathematics, Tianjin University, China. E-mail: {yl7268, qingyin, wuou}@tju.edu.cn, llhou@mail.nankai.edu.cn

Jie Gui is with the Department of Computational Medicine and Bioinformatics, University of Michigan, USA. E-mail: guijie@ustc.edu

James Kwok is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, China. E-mail: jamesk@ust.hk

[1]https://www.facebook.com/pg/jobbot.me/about/?ref=page_internal.

[2]To our knowledge, only our early work [4] explored this issue. This study is an extension of our early work [4]. Nevertheless, a larger data corpus is compiled and a whole new deep neural network is proposed.

A large data corpus[3] is constructed to facilitate the investigation of answer understanding in reverse QA. The experimental results indicate that the AntNet is significantly better than the competing methods.

Our contributions are summarized as follows:

- An under-explored natural language understanding task, namely, answer understanding in reverse QA, is investigated. To our knowledge, this is the first work that focuses on this task.
- A new data corpus is compiled and publicly available for interesting readers.
- A novel AntNet is proposed. The network consists of three key modules, including skeleton extraction for questions, relevance-aware representation of answers, and multi-hop based fusion and significantly outperforms existing methods in the experiments.

II. Related Work

QA and Reverse QA are firstly reviewed. The NLP for answer analysis is reviewed latter as this study also aims to analyze answer texts. Our proposed methodology is inspired by aspect-based sentiment analysis (ABSA), so ABSA is also reviewed.

A. Question answering (QA)

QA is a crucial task that depends on natural language understanding and domain knowledge [3]. QA aims to return an appropriate answer to a user's question. The answer is usually selected from an answer corpus on the basis of a question-answer matching model. The model calculates the matching score between the question and each candidate answer. The answer with the highest matching score is then used to return to users.

In traditional QA methods, features of questions and answers are extracted by conventional methods, like tf-idf [8], lexical cues [9], word order [10] and so on. Thereafter, a similarity scoring function, such as cosine, is used to calculate the matching score.

In deep QA methods, features of questions and answers are extracted by deep learning methods, like convolutional neural network (CNN) [11], LSTM [12], or Transformer [7]. An end-to-end framework is usually used to combine the deep feature extraction and the successive matching function training [13], [14].

B. Reverse QA

In addition to meet users' information requirements, machines in some real applications, such as telephone survey and commercial intelligent customer service systems, are also required to actively acquire the exact needs or feedbacks of users [15]. Accordingly, machines may choose to proactively raise questions to users and then analyze their answers. In other words, machines are the questioners and humans are the answerers. This process is a reverse of the conventional QA process and is called reverse QA in this paper. Fig. 2 shows the conventional QA and reverse QA processes.

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

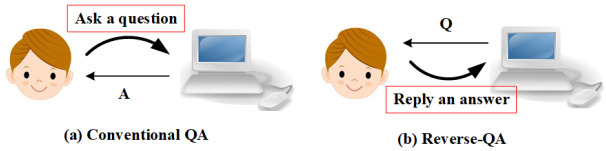

Fig. 2. The difference between conventional QA (a) and reverse QA (b).

In fact, although reverse QA is not explicitly described in previous literature, it has been referred in various studies especially in human-machine dialog [16]. Liu et al. [17] utilized a bidirectional LSTM (BiLSTM) to encode machine's questions and user's answers to a continuous representation in an end-to-end task oriented dialogue system. Similarly, Zhang et al. [18] take a question-answer pair as the input of slot-value memory and external memory module. Furthermore, Westion [19] improved machine's ability to learn from the human's feedback by introducing forward prediction (FP) that learns by predicting textual feedback.

C. NLP for answer texts

Existing NLP studies for answer texts are mainly about question generation and answer retrieval.

In question generation, early work tackled question generation with a rule-based approach [20] or an overgenerate-and-rank approach [21] which relied heavily on well-designed rules or manually crafted features respectively. To overcome these limitations, Du et al. [22] introduced a deep sequence-to-sequence learning approach to generate questions. Rao et al. [23] introduced Generative Adversarial Networks (GANs) to generate questions that are more useful and specific to the context.

Compared with typical document retrieval, the answer retrieval model needs to exploit more semantic information. Inspired by the advantage of translation in modeling the relationship between words, Xue et al. [24] used a translation-based approach to solve the problem of mismatching. Subsequently, popular neural networks like CNN [25] and LSTM [26] were used in this task. Tay et al. [27] proposed a recurrent network using temporal gates to learn interactions between question-answer pairs.

In this study, answer understanding is transformed into an answer classification task. Fig. 3 shows the main difference between answer retrieval in QA and answer classification in reverse QA. In QA, answer selection relies on a matching model between a given human question and candidate answers, whereas, in reverse QA, answer classification relies on a model that classifies the answer into one of the predefined categories. The difference between answer-processing tasks for QA and reverse QA is quite evident.

Our early proposed network, semi-interactive attention network (Semi-IAN) [4], is based on an ABSA network called interactive attention network (IAN). A data corpus is compiled to verify the effectiveness of Semi-IAN. On the basis of this early work, this study compiles a larger data corpus and proposes a more effective network to capture the dependency between machine questions and human answers.

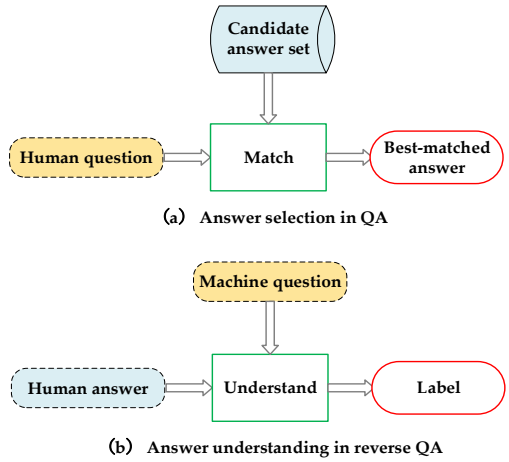

Fig. 3. The main difference between answer selection in QA and answer understanding in reverse QA.

D. Aspect-based sentiment analysis (ABSA)

ABSA is also briefly reviewed as our proposed method is partially motivated by ABSA networks.

Considering a sentence and an aspect (or an aspect term), ABSA aims to predict the sentiment polarity for the given aspect [28]. ABSA methods have two typical categories. The first category integrates recurrent neural networks (RNN) with attention mechanism [29], and the second category utilizes a memory network with a multi-hop attention [5]. Recently, researchers have introduced new advancements in NLP to ABSA, such as Transformer and temporal convolutional network (TCN) [30]. These new models are mainly used to infer additional effective representations of sentences and aspects.

III. Problem and Data

We first give an analysis for answer understanding in reverse QA as it is rarely investigated.

A. Problem analysis

The primary difficulty in answer understanding results from the openness of the corresponding question. For instance, three machine questions are given as follows:

- MQ1: Do you like sports?
- MQ2: Which sport do you like best, swimming, climbing, or playing football?
- MQ3: Which sport do you like?

The difficulty in answer understanding for the three questions is very likely to increase. The answers for MQ3 will be quite difficult to understand considering the following answer examples: (1) "It depends on the weather", (2) "Competitive sports", and (3) "Water sports".

MQ1 is a true/false (T/F) question; MQ2 is a multi-choice (MC) question; and MQ3 is nearly an open question. Intuitively, open questions raise the naturalness of the interaction between machines and humans. However, it is quite challenging for the answer understanding of open questions.

This study considers the first two types, namely, T/F and MC questions. Consequently, answer understanding becomes a classification problem. One may argue that a simple regularized matching strategy can deal with the classification problem. The simple regularized matching strategy is not an ideal solution due to the following reasons:

- The regularized matching strategy has low generation performance. This method relies on key words, such as 'ok', 'yes', and 'not'. These key words directly indicate a 'true' or 'false' answer. Take the question "Are you a teacher?" as an example. A simple regularized matching strategy may considers the answer "I am a police" means 'true' if the regularization term is "I am".
- The compiling of matching rules for different questions is a tedious job. As simple rules lead to high error rates of understanding, the matching rules are required to be highly complex. As a consequence, the compiling of an effective rule set is quite difficult especially for MC questions.

In our experiments, the simple regularized matching strategy is tested on the T/F questions. The accuracy is low[4], indicating that a machine learning-based approach should be investigated.

The successive subsection gives a formal description.

B. Problem formulization

Let O be the option term set for a question. In MC questions, O is defined as the set of concrete option terms. For instance, O is defined as $\{$*'swimming', 'climbing', 'playing football'*$\}$ for MQ2; in T/F questions, O is defined as $\{$*'Yes'*$\}$ to ensure consistency with the format of MC questions.

We first illuminate how answer understanding is transformed into answer classification with concrete examples. The (answer) label set L is defined as $\{$*'true', 'false', 'uncertain'*$\}$. Let q_i be the question and $o_{i,k}$ be the k-th option term of q_i. For MQ1, given an answer $s_{i,j}$ for question q_i, answer understanding equals to classify $\{q_i, s_{i,j}, o_{i,k}\}$ into one of the labels in the set L. $o_{i,k}$ ($o_{i,k} \in O$) is *'Yes'* here. For MQ2, given an answer $s_{i,j}$, answer understanding equals to three sub-classification tasks, i.e., the classification of $\{q_i, s_{i,j}$, *'swimming'*$\}$, $\{q_i, s_{i,j}$, *'climbing'*$\}$, and $\{q_i, s_{i,j}$, *'playing football'*$\}$ into one of the labels in the label set L.

The answer classification for T/F and MC questions can be further formalized as follows:

We aim to predict the category $l_{i,j,k}$ ($l_{i,j,k} \in L$) of the triplet $\{q_i, s_{i,j}, o_{i,k}\}$ by considering the machine-question and human-answer pair $\{q_i, s_{i,j}\}$, the corresponding option term $o_{i,k}$ of the question, and a predefined answer-label set L.

The number of option terms is only one (as $O = \{$*'Yes'*$\}$) in T/F questions. Therefore, o in the triplet can be omitted in such question type.

C. Data construction

Existing QA and text classification benchmark data sets are inappropriate for training and evaluating reverse QA models.

[4]The regularized matching rules (in Chinese) are attached in the supplementary materials for interested readers.

TABLE I
STATISTICS OF TDATA AND MDATA

Data	questions	answers	samples (true/false/uncertain)
TData	536	10,817	4,452/4,610/1,755
MData	517	32,511	15,000/10,540/6,971

Therefore, two data sets are compiled with a standard labeling process. The MC questions we studied is limited in the type that the options appear in the question, which we call option-contained MC questions.

For the two data sets, the questions are constructed as follows. First, seven domains are selected, namely, encyclopedia, insurance, personal, purchases, leisure interests, medical health, and exercise. A total of twenty graduate students, specifically ten males and ten females, were invited to participate in the data compiling using Email advertising from our laboratory. All the participants are Chinese and range in age from 22 to 30. Considering that the question and answer generations are not difficult to understand, we did not give special instructions to the participants. Each participant was allowed to construct 50 to 60 questions. Finally, 1053 questions are obtained after deleting some invalid questions. Among that, the numbers of T/F and MC questions are 536 and 517, respectively.

The questions were equally and randomly assigned to the twenty participants. Each question was given 18 to 22 answers. The participants also labeled the answers generated by themselves considering that the other participants didn't know what exactly the answer means. A new data corpus was obtained. Table I shows the details. The data corpus contains two data sets, namely, TData and MData.

For the TData, the types of answers are roughly divided into affirmative, negative, uncertain, and unrelated. Given that the uncertain and unrelated answers are similar in function to the question, we classify them as the same class. In this way, each answer is tagged with '1' for true, '0' for false, and '2' for uncertain or unrelated. Each sample consists of three components: question, answer, and label. The total number of samples is 10,817.

For the MData, the number of options for each MC question are different and cannot be categorized uniformly. Thus, we add the option information to the MC questions and get a series of transformed MC questions as described in Section III-B. Therefore, the same answer to the same question will have different labels for dissimilar options. Similarly, '1' indicates that the answer is a 'true' answer to the current option, '0' implies that the answer is a 'false' answer to the current option, and '2' denotes that the answer is a 'uncertain' answer to the current option or the answer is meaningless to this question. Each sample consists of four components: question, option, answer, and label. There are 32,511 transformed MC training samples.

IV. METHODOLOGY

Section III-B describes that answer understanding is transformed into an answer classification problem. Obtaining the deep representations of the machine-question and human-answer pair and a given option term is the first step. In addition, questions provide the context for answer understanding. The final dense representation should consider the contextual dependency between questions and answers.

Inspired by related research in text classification and aspect-based sentiment analysis, we propose a new deep model, called AntNet. Fig. 4 shows the main structure.

The experimental data are in Chinese. Therefore, the word means "the Chinese word" in the following subsections.

The input of the AntNet is the triplet $\{q_i, s_{i,j}, o_{i,k}\}$. $o_{i,k}$ is indicated by an indicator vector. The indicator is set as a zero vector for all samples in T/F questions, and the option indicator is set as a one-hot vector in MC questions. The left part of the AntNet deals with the input of q_i and $o_{i,k}$ to generate two representations. The first representation characterizes the mixture of q_i and $o_{i,k}$, while the second representation characterizes important information, which is called skeleton (Chinese) words for questions in this study. The first representation is called full representation, while the second is called skeleton representation.

The lower-right portion deals with the input of answers, and the output is a set of hidden dense vectors for answers. In this part, a relevance-aware module is used to well characterize the relevance cues contained in the answers considering that users may return irrelevant texts.

The upper-right portion deals with the contextual dependency between questions and answers to obtain an overall dense feature vector, which is fed into the final decision softmax layer. A multi-hop attention mechanism is used in this part.

The following subsections introduce the details of the three parts. The skeleton information extraction which is firstly introduced.

A. Unsupervised skeleton extraction for questions

Question texts usually contain redundant[5] or even disturbed words, which may negatively influence answer understanding. The skeleton information in a question should be extracted. Skeleton information refers to the words which directly affect how users respond to.

Intuitively, skeleton information extraction can be performed a supervised manner. Alternatively, the skeleton words are manually labeled for a set of training text samples. These training samples are then fed into a sequence labeling model for training. The trained sequence labeling model can be used to extract skeleton words for new texts. Nevertheless, as it is difficult to provide an explicit and formal definition for skeleton words, it is thus also difficult for human labeling. To this end, an unsupervised extraction manner is proposed in this study.

A training sample is a triplet $\{q_i, s_{i,j}, o_{i,k}\}$ in this study, where q_i is the i-th question, $s_{i,j}$ is the j-th answer for q_i,

[5]These words may be used for increasing the interaction interestingness.

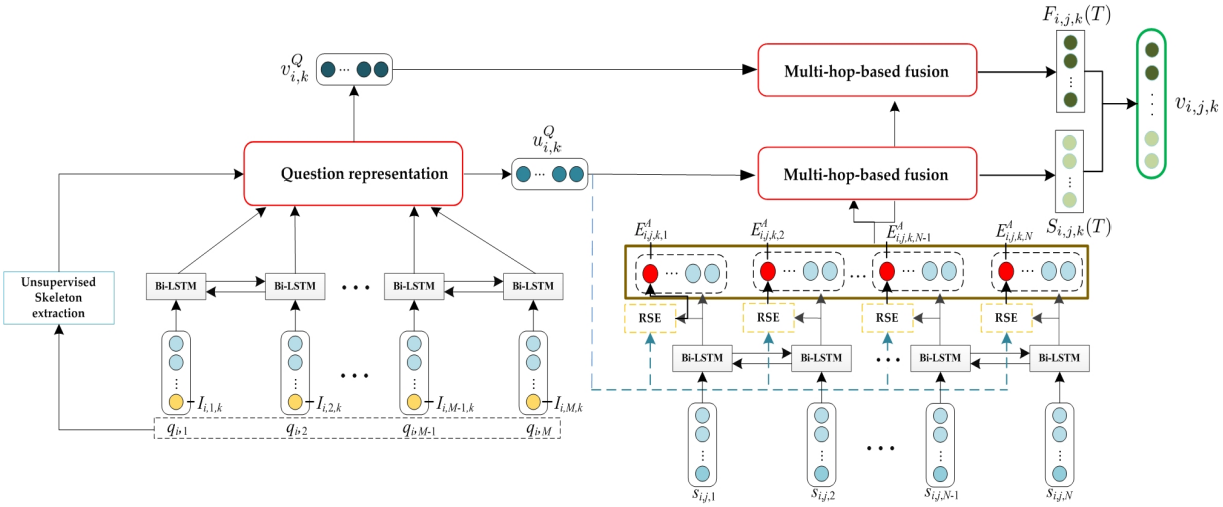

Fig. 4. The structure of AntNet.

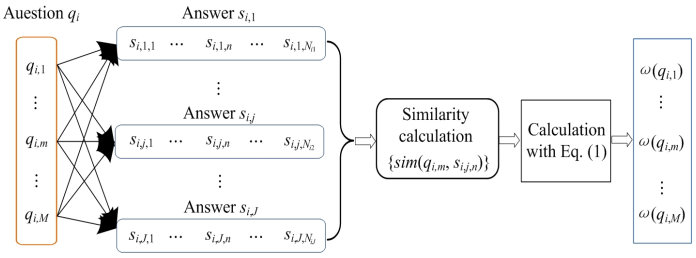

Fig. 5. Skeleton score calculation pipeline.

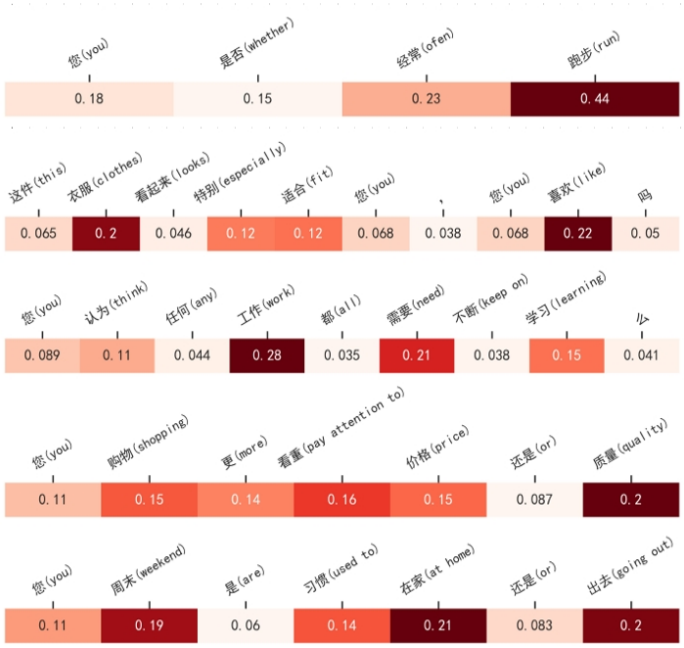

Fig. 6. Skeleton scores for words in five questions. All the data in this study are in Chinese. To facilitate English readers, the Chinese words in the above questions are translated into English.

and $o_{i,k}$ is the k-th option term for q_i. The primary difference between this study and conventional classification studies lies in that many training samples in this study share the same element 'q_i'. In other words, each question (q_i) corresponds to multiple answers $(s_{i,1}, \cdots, s_{i,j}, \cdots, s_{i,J})$, leading that there are multiple training samples for q_i and a fixed option term $o_{i,k}$ including $\{q_i, s_{i,1}, o_{i,k}\}, \cdots, \{q_i, s_{i,J}, o_{i,k}\}$. Let $q_i = \{q_{i,1}, \cdots, q_{i,m}, \cdots, q_{i,M_i}\}$ be the i-th question, where M_i is the word-level length of the question and $q_{i,m}$ is the m-th word of q_i. Let $s_{i,j} = \{s_{i,j,1}, \cdots, s_{i,j,n}, \cdots, s_{i,j,N_{ij}}\}$ be the j-th answer for q_i, where N_{ij} is the word-level length of the answer and $s_{i,j,n}$ is the n-th word. A score can be calculated for $q_{i,m}$ as follows.

$$\omega(q_{i,m}) = \frac{1}{J}\sum_{j=1}^{J} \max\{sim(q_{i,m}, s_{i,j,n}), n = 1, ..., N_{ij}\} \quad (1)$$

where $sim(,)$ calculates the similarity of two words according to their word embeddings.

The score calculation pipeline is shown in Fig. 5. The question words whose scores are higher than a threshold are considered skeleton words in this study. Let Sk_i be the set of skeleton words for q_i. The threshold is empirically set as 0.12 according to our experimental evaluation.

Fig. 6 shows several questions and their associated scores on each Chinese word calculated by Eq. (1). The words with scores higher than 0.12 are key words in their corresponding questions. The scores of the words such as 'you', 'are', 'this', 'or' are low in most sentences. The words which are directly related user options such as 'running', 'like', 'quality', 'at home' have higher scores.

B. Question representation

As stated earlier, two-level representations are considered in the AntNet.

The first-level representation (referred to skeleton representation) characterizes the skeleton words in the question, while the second-level representation (referred to full representation) characterizes the whole question. The two representation vectors are calculated as follows.

The given training sample is represented by an input triplet $\{q_i, s_{i,j}, o_{i,k}\}$ and its label $l_{i,j,k}$. Let $I_{i,m,k}$ be an indication vector for whether the word $q_{i,m}$ is in $o_{i,k}$[6]. If $q_{i,m}$ is in $o_{i,k}$, $I_{i,m,k} = 1$; otherwise $I_{i,m,k} = 0$.

After the encoding of a BiLSTM on q_i, the hidden representation of each word is defined as following:

$$h^Q_{i,m,k} = BiLSTM_Q(h^Q_{i,m-1,k}, h^Q_{i,m+1,k}, q_{i,m}, I_{i,m,k}) \quad (2)$$

where $h^Q_{i,m,k} \in \mathbb{R}^d$ and $q_{i,m}$ represents embedding vector of words.

With the extracted skeleton words and their associated scores calculated by Eq. (1), the skeleton representation for a question q_i (given $o_{i,k}$) is calculated as following:

$$u^Q_{i,k} = \sum_{q_{i,m} \in Sk_i} \omega(q_{i,m}) h^Q_{i,m,k} / \sum_{q_{i,m} \in Sk_i} \omega(q_{i,m}). \quad (3)$$

The full representation $v^Q_{i,k}$ of q_i (given the invovled option term $o_{i,k}$) is calculated on the basis of attention scores $\{att^Q_{i,m,k}\}_{m=1}^{M_i}$ for each word $q_{i,m}$. The calculation is described as follows:

$$\begin{aligned} a^Q_{i,m,k} &= {h^Q_{i,m,k}}^T W_a u^Q_{i,k} \\ att^Q_{i,m,k} &= \frac{exp(a^Q_{i,m,k})}{\sum_{m=1}^{M_i} exp(a^Q_{i,m,k})} \\ v^Q_{i,k} &= \sum_{m=1}^{M_i} att^Q_{i,m,k} h^Q_{i,m,k} \end{aligned} \quad (4)$$

where $W_a \in \mathbb{R}^{d \times d}$, $a^Q_{i,m,k}, att^Q_{i,m,k} \in \mathbb{R}$, and $v^Q_{i,k} \in \mathbb{R}^d$.

C. Relevance-aware answer representation

BiLSTM is also utilized to generate the hidden vectors of answer texts with the following calculation:

$$h^A_{i,j,n} = BiLSTM_A(h^A_{i,j,n-1}, h^A_{i,j,n+1}, s_{i,j,n}) \quad (5)$$

where $h^A_{i,j,n} \in \mathbb{R}^d$.

To maintain the naturalness of the whole interaction, users can return their answers in arbitrary forms and with arbitrary contents. Therefore, some irrelevant texts are included in some answers even if these answers do not belong to the 'irrelevant' category. Thereafter, a relevance score is calculated for each word ($s_{i,j,n}$) in the answer texts ($s_{i,j}$) with the following equation:

$$p^A_{i,j,k,n} = sigmoid(W_p[h^A_{i,j,n}, u^Q_{i,k}] + b_p). \quad (6)$$

The length of $p^A_{i,j,k,n}$ is one which is much smaller than $h^A_{i,j,n}$ in our practical implementation. Consequently, the proportion of the $p^A_{i,j,n}$ part is quite small in the concatenated vectors, which limits the advantages of the relevance vectors. We adopt the trick used in [31] in our implementation. The length of $p^A_{i,j,n}$ is enlarged according to:

$$E^A_{i,j,k,n} = p^A_{i,j,k,n} \otimes 1_{N_e \times 1} \quad (7)$$

where $1_{N_e \times 1}$ is an N_e-dimensional vector. $E^A_{i,j,k,n}$ is the enlarged vector. Parameter N_e is used to increase the length of $p^A_{i,j,k,n}$. Fig. 7 shows the steps of **r**elevance **s**core calculation and dimensionality **e**nlarging (RSE). Experimental results validate the effectiveness of the dimensionality increment for $p^A_{i,j,k,n}$.

The relevance score vector is concatenated with the hidden vectors for each word:

$$h'^A_{i,j,k,n} = \begin{bmatrix} h^A_{i,j,n} \\ E^A_{i,j,k,n} \end{bmatrix} \quad (8)$$

where $h'^A_{i,j,k,n} \in \mathbb{R}^{d+N_e}$ is the updated hidden representation of each answer word.

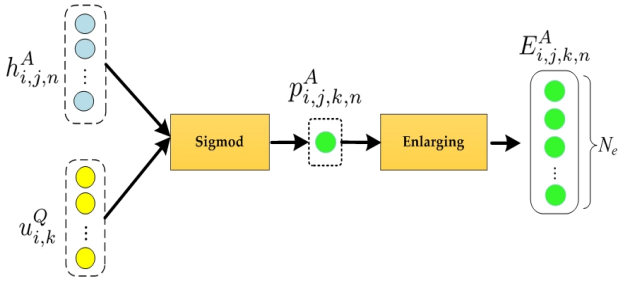

Fig. 7. The **r**elevance **s**core calculation and dimension **e**nlarging (RSE).

D. Multi-hop based fusion

The representations (i.e., $u^Q_{i,k}, v^Q_{i,k}, h'^A_{i,j,k}$) are fused to obtain the final representation of the whole triplet $\{q_i, s_{i,j}, o_{i,k}\}$.

Inspired by the ABSA [28], a multi-hop based question-answer fusion module is introduced. This module can well represent the input machine-question and human-answer pair and the associated option term.

The vectors $u^Q_{i,k}$ and $v^Q_{i,k}$ are seperately input into the multi-hop based fusion module. Fig. 8 shows the multi-hop based fusion. The left part and the right part are the iterative approaches for $v^Q_{i,k}, u^Q_{i,k}$ given $h'^A_{i,j,k}$, respectively.

The calculation with $v^Q_{i,k}$ and $h'^A_{i,j,k}$ is taken as an example. Let $F_{i,j,k}(0) = v^Q_{i,k}$ be the input question representation. The first hop (hop 1 in Fig. 8) is computed as following:

$$\begin{aligned} F_{i,j,k}(0) &= v^Q_{i,k} \\ m^{(1)}_{i,j,k,n} &= W^{(1)}_m tahn(W^{(1)}_h h'^A_{i,j,k,n} + W^{(1)}_x F_{i,j,k}(0) + b^{(1)}) \\ a^{(1)}_n &= \frac{exp(m^{(1)}_{i,j,k,n})}{\sum_{n=1}^{N_{i,j}} exp(m^{(1)}_{i,j,k,n})} \\ x' &= \sum_{n=1}^{N_{i,j}} a^{(1)}_n h'^A_{i,j,k,n}. \end{aligned} \quad (9)$$

An active module is used to obtain a new vector:

$$F_{i,j,k}(1) = tanh(W_{f1} x' + b_f) + W_{f2} F_{i,j,k}(0) \quad (10)$$

where $F_{i,j,k}(1)$ is also the input of the second hop (hop 2 in Fig. 8).

The above step is iterated T times to obtain the feature vector $F_{i,j,k}(T)$.

Finally, $F_{i,j,k}(T)$ from full representation $v^Q_{i,k}$ and $S_{i,j,k}(T)$ from skeleton question representation $u^Q_{i,k}$ are concatenated into one representation vector:

$$v_{i,j,k} = \begin{bmatrix} F_{i,j,k}(T) \\ S_{i,j,k}(T) \end{bmatrix}. \quad (11)$$

[6]Some option terms are phrases.

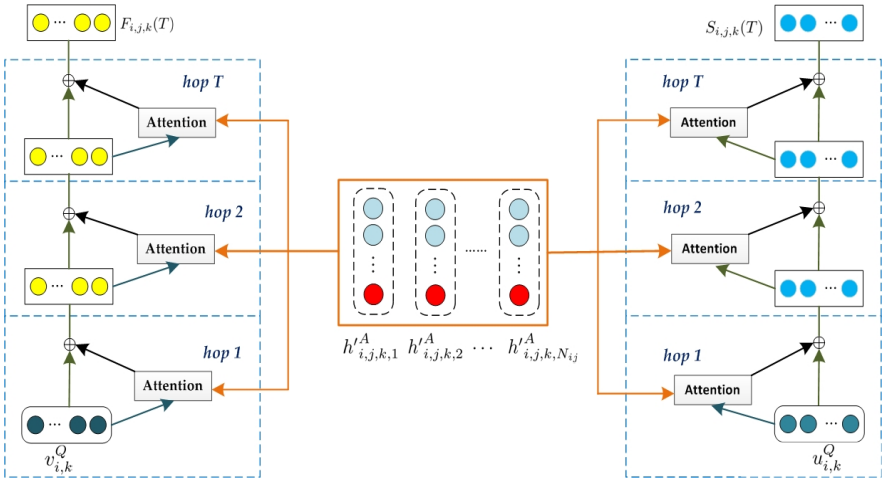

Fig. 8. Multi-hop based fusion for the involved question (two feature vectors $u^Q_{i,k}$ and $v^Q_{i,k}$) and answer ($h'^A_{i,j,k}$).

The predicted label is calculated as following:

$$l'_{i,j,k} = softmax(W v_{i,j,k} + b). \tag{12}$$

With the predicted and ground truth labels, the AntNet can be learned with the following cross entropy loss function:

$$loss = -\sum_{i,j,k} l_{i,j,k} log l'_{i,j,k}. \tag{13}$$

E. Algorithmic steps

Algorithm 1 shows the algorithmic steps of the entire learning procedure for the proposed AntNet. Because the skeleton words are seperately extracted, the overall training algorithm is not an end-to-end procedure. Accordingly, Algorithm 1 contains two main parts. In the first part, the skeleton words for each training question are extracted with an unsupervised manner. In the second part, the whole network AntNet is trained with the supervised labels.

Algorithm 1: Learning for AntNet

Input: Training set (T), Word2Vec for each Chinese words, threshold τ for skeleton words, hyper-parameters (e.g., learning rate and dropout rate).

Output: A trained AntNet.
Steps:

1. Construct the relevant answer set for each question.
2. Calculate the scores ω for words in each question according to Eq. (1).
3. Generate the skeleton words for each question according to ω and the threshold τ.
4. Generate the indicator for the specified opinion term of each question.
5. Train the AntNet with T on the basis of the input constructed in Steps 3, 4, and the loss function defined in Eq. (13).

V. Experiments

This section shows the evaluation of the proposed AntNet in terms of the whole network and the three key modules, namely, skeleton extraction for questions, relevance-aware representation of answers and multi-hop based fusion.

A. Competing methods

Several classical and state-of-the-art deep model-based algorithms are used and listed as follows:

- **BiLSTM (A)**: The standard BiLSTM is used to encode the answer texts directly, and the dense vector is used for answer classification.
- **BiLSTM (Q+A)**: The standard BiLSTM is also used for both question and answer texts.
- **RAM** [5]: It leverages the hidden vectors of BiLSTM as memory vectors. Then, a GRU is used to construct a multi-hop based fusion for memory vectors and input target vector. The final dense vector contains information from both sentences and targets. This study takes question texts as target texts.
- **ATAE** [29]: ATAE is based on BiLSTM and proposed for target-based sentiment analysis. The target vector is concatenated with the word embedding of each word. In this experiment, the question texts are taken as the target texts.
- **Transformer (A)**: The standard Transformer is used to encode the answer texts directly and the averaging pooling of the hidden vectors of the last layer is used for answer classification.
- **Transformer (Q+A)**: Both questions and answers are concatenated and input into the standard transformer.
- **Semi-IAN** [4]: It is the first method related to answer understanding in reverse QA. The interaction between question and answers are modeled.
- **Regularized Matching (RM)**: This method is an engi-

neering solution which matches pre-defined key words or phrases.

Our proposed method consists of several new modules. To investigate the validity of three major components: skeleton extraction, relevance-aware answer representation, and multi-hop based fusion, we test AntNet with or without these components. The variants of our method are listed as follows:

- **AntNet w/o SR**: The AntNet without the skeleton representation.
- **AntNet w/o RR**: The AntNet without the relevance-aware representation.
- **AntNet w/o MF**: The AntNet without the multi-hop based fusion.
- **AntNet**: The entire AntNet with all introduced key components.

As the answer understanding for reverse QA is investigated from a classification perspective, the classification accuracy is used as the performance metrics.

B. Training setting

Two data corpora, namely, TData and MData, are involved in our experiment. They are divided according to the following rules:

(1) Each data corpus is divided into two parts with the proportion 4:1. Four folds are used for training, and the rest is used for testing.

(2) A proportion of 10% samples in training data are used as validation data.

We use 256-dimension Word2Vector embeddings trained on our own corpus. We initialize the word embeddings randomly for the out-of-vocabulary words. In addition, the lengths of questions and answers are truncated as 33. The learning rate and dropout ratio are set to 5×10^{-4} and 0.2, respectively. We minimize the loss function using the ADAM optimizer [32] and fix the word embedding vectors during the training. Furthermore, we set the remaining parameters as default values.

All above mentioned models are trained with Tensorflow.

C. Overall competing results

Table II presents the main results (classification accuracies) of the competing methods on two data corpora. AntNet achieves the highest accuracies on both data corpora. Compared with the state-of-the-art network, Transformer, the results significantly increased. The poor performance of Transformer may result from the small training size.

The existing answer understanding method, Semi-IAN, is inferior to RAM. In fact, Semi-IAN is a slight variation of the ABSA network, IAN. As RAM is also an ABSA method, RAM unsurprisingly outperforms Semi-IAN. Among these methods, the RM method has the lowest accuracy of 50.38%. Hence, a machine learning-based approach is essential.

D. Evaluation of the different modules of AntNet

In this subsection, we verify the usefulness of the three introduced key modules, namely, skeleton representation of questions, relevance-aware representation of answers, and multi-hop based fusion. The involved competing methods are AntNet w/o SR, AntNet w/o RR, AntNet w/o MF, and the entire network AntNet.

TABLE II
ACCURACIES ON TDATA AND MDATA.

Method	Accuracy (TData)	Accuracy (MData)
BiLSTM (A)	0.7375	0.6878
BiLSTM (Q+A)	0.7196	0.6905
RAM	0.7503	0.7121
ATAE	0.7458	0.6865
Transformer (A)	0.7029	0.6547
Transformer (Q+A)	0.5830	0.6167
Semi-IAN	0.7485	0.6986
AntNet	**0.7921**	**0.8213**

TABLE III
RESULTS OF ANTNET AND ITS VARIATIONS (WITHOUT CERTAIN KEY MODULES) ON TDATA AND MDATA.

Method	Accuracy (TData)	Accuracy (MData)
AntNet w/o SR	0.7853	0.7931
AntNet w/o RR	0.7536	0.7432
AntNet w/o MF	0.7858	0.7896
AntNet	**0.7921**	**0.8213**

Table III shows the competing results on the two data corpora, TData and MData, respectively. Unsurprisingly, all variations without a certain type of key module achieve inferior accuracies compared with the full version of AntNet. These comparisons indicate that all the three key modules are useful in answer understanding.

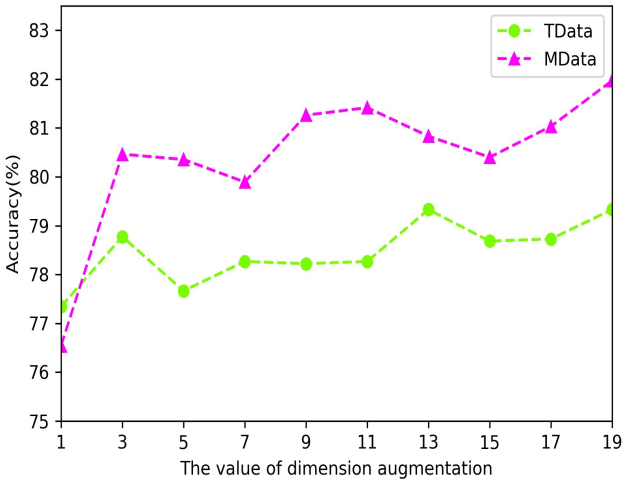

Fig. 9. Understanding accuracies under the different values of dimension augmentation for relevance-aware representation.

The comparison of the three variations shows that AntNet w/o RR (AntNet without the relevance-aware representation) obtains the lowest accuracies. On MData, the accuracy achieved by AntNet w/o RR is roughly 7% lower than that

TABLE IV
EXAMPLES IN WHICH HUMAN ANSWERS CONTAIN IMPLICIT RELEVANCE HINTS.

Machine question	Human answer	Labels
你是喜欢裙子还是裤子? (Do you like skirts or pants?)	我不挑. (I am not picky.)	'True' for both 'skirts' and 'pants'.
您周末喜欢逛街还是打游戏? (Do you like shopping or playing games on weekends?)	我是女生哎! (I'm a girl.)	'True' for 'shopping' and 'False' for 'playing games'.
您平时喜欢喝热水还是凉水? (Do you like hot or cold water?)	我爱喝苏打水. (I like to drink soda.)	'False' for both 'hot' and 'cold water'.
您习惯晨跑还是夜跑? (Are you used to running in the morning or at night?)	我喜欢看别人跑. (I like to watch others run.)	'False' for both 'in the morning' and 'at night'.

by the AntNet. These results reflect that the relevance-aware representation module is essential for the whole network.

In the relevance-aware representation, the dimension of the relevance score is augmented by using Eq. (7). We perform an experiment to investigate the performances of the AntNet under different augment parameters N_e in Eq. (7). Fig. 9 shows the accuracies of the AntNet according to different N_e values. With the increase of the value of N_e, the understanding accuracies on both sets demonstrate an increasing trend. When the value equals 19, the AntNet achieves the maximum accuracies on both data sets.

In the multi-hop module, the number of hops is also an important parameter. We also perform experiments to explore the relationship between hop count and final accuracy. Fig. 10 shows the accuracies of the AntNet under the different numbers of hops.

The number of hops also influences the final performance. The highest value (when the number equals 4) is nearly 3% higher than the lowest value (when the number equals 7) on TData. On MData, the overall trend increases, and the accuracy is the highest when the number equals 10.

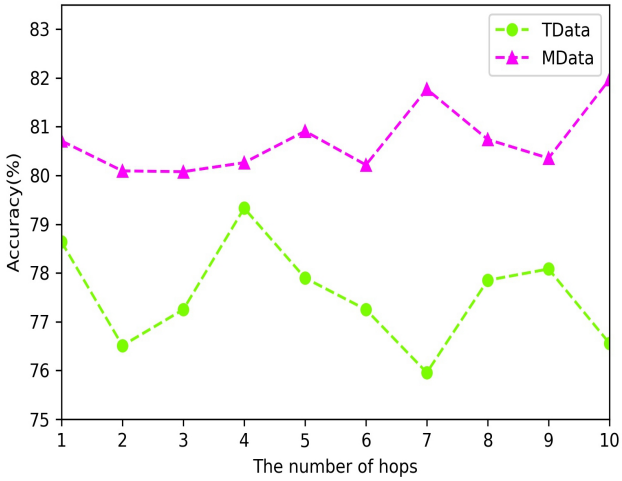

Fig. 10. Accuracies under the different numbers of hops for multi-hop based fusion.

E. Discussion

We empirically analyze the error understanding answers in the test set to well analyze the performance of the AntNet. The results show that the errors are prone to occur for answers containing implicit preference information. Particularly, once the implicit information contains negative or positive words, they are very likely to be error judged. Table IV shows several examples of answers containing implicit information. The first question belongs to the MC type, so each label should corresponding to an option term such as 'skirts' and 'pants'.

The fourth question-answer pair is taken as an example. The answer does not provide a direct reply toward the question. In fact, the answer means that the user does not like to run in the morning or night neither. The future work will focus on the extraction of additional hints for users' choices.

Attention is the core of deep neural networks in NLP [7]. The following example is visualized to facilitate the analysis of the effectiveness of the multi-hop attention used in this study.

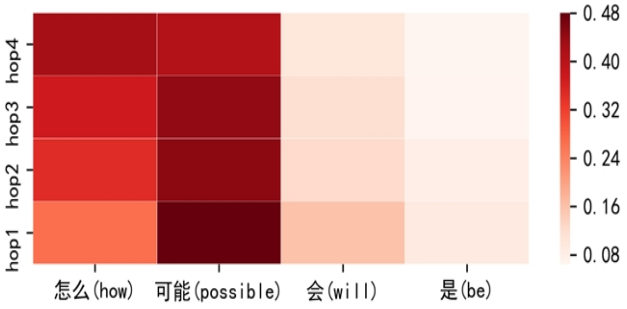

Fig. 11. Multi-hop attention values for an answer sentence.

In hop1 shown in Fig. 11, the attention scores for the Chinese word '怎么' (how) is quite small. Nevertheless, in hop4, their attention scores become high, which is reasonable because the Chinese word is quite important for answer

understanding.

We also investigated the relationship between training data and model performances. Fig. 12 shows the variations of performances under different proportions of training data on TData. With the increasing of training data, the performances of the three methods, i.e., AntNet, BiLSTM, and semiIAN, are also increased. Nevertheless, when the training data is small, the performance of AntNet is also relatively good. Similar observations are obtained on MData.

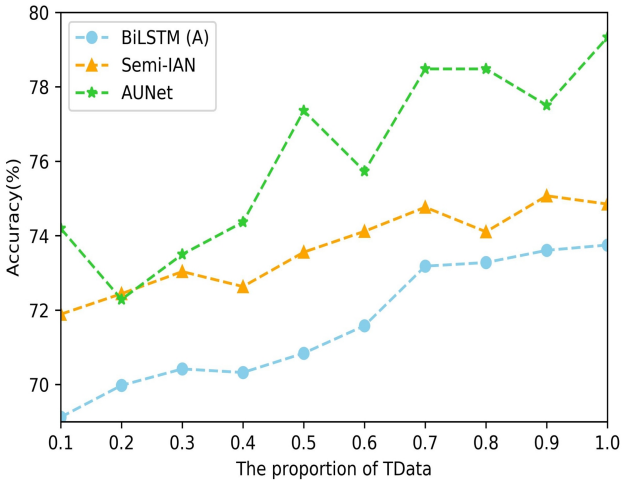

Fig. 12. Accuracies under the different proportions of training data on TData.

VI. Conclusion

The automatic understanding of human answers in reverse QA can bring natural interactions and thus improve user experiences. However, it receives little attention in previous literature. This study compiles a relatively large data corpus for answer understanding in reverse QA. An effective deep neural network, called AntNet, is proposed for the understanding of the answers for the two most common types of questions.

The AntNet utilizes two types of questions and a relevance-aware presentation for answer texts. The multi-hop based fusion module is used for modeling the contextual dependency between questions and answers. The experimental results indicate that the AntNet is significantly better than the exiting method and state-of-the-art NLP models with direct variations.

Acknowledgement

We thank Dr. Guan Luo, Dr. Xiaodong Zhu, and Prof. Qinghua Hu for their contributions in our early proposed model, Semi-IAN in PAKDD paper.

References

[1] B. Hixon, P. Clark, and H. Hajishirzi, "Learning knowledge graphs for question answering through conversational dialog." in *NAACL-HLT*, 2015, pp. 851–861.

[2] A. Kumar, O. Irsoy, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing." in *ICML*, 2016, pp. 1378–1387.

[3] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering." *CoRR*, vol. abs/1611.01604, 2016.

[4] Q. Yin, G. Luo, X. Zhu, Q. Hua, and O. Wu, "Semi-interactive attention network for answer understanding in reverse-qa." in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, vol. 11440, 2019, pp. 3–15.

[5] P. Chen, Z. Sun, L. Bing, and 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.

[6] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM." in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 5876–5883.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need." in *Advances in neural information processing systems 30*, 2017, pp. 5998–6008.

[8] G. Salton and C. T. Yu, "On the construction of effective vocabularies for information retrieval." vol. 9, no. 3, pp. 48–60, 1973.

[9] K. Khalifa and N. Omar, "A hybrid method using lexicon-based approach and naive bayes classifier for arabic opinion question answering." vol. 10, no. 10, pp. 1961–1968, 2014.

[10] E. Hovy, U. Hermjakob, and C. Y. Lin, "The use of external knowledge in factoid QA," in *In Proceedings of the Tenth Text REtrieval Conference*, 2001, pp. 644–652.

[11] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, "Improved representation learning for question answer matching." in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 464–473.

[12] Y. Tay, M. C. Phan, L. A. Tuan, and S. C. Hui, "Learning to rank question answer pairs with holographic dual lstm architecture." in *ACM SIGIR*, 2017, pp. 695–704.

[13] Z. Wang, W. Hamza, and R. Florian., "Bilateral multi-perspective matching for natural language sentences." *CoRR*, vol. abs/1702.03814, 2017.

[14] A. W. Yu, D. Dohan, M. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "Qanet: Combining local convolution with global self-attention for reading comprehension." *CoRR*, vol. abs/1804.09541, 2018.

[15] U. Krcadinac, J. Jovanovic, V. Devedzic, and P. Pasquier, "Textual affect communication and evocation using abstract generative visuals." *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 370–379, 2015.

[16] A. Preece, W. Webberley, D. Braines, E. G. Zaroukian, and J. Z. Bakdash, "Sherlock: Experimental evaluation of a conversational agent for mobile information tasks." *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 6, pp. 1017–1028, 2017.

[17] B. Liu, G. Tür, D. Hakkani-Tür, P. Shah, and L. Heck, "Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems." in *Proceedings of NAACL: Human Language Technologies*, 2018, pp. 2060–2069.

[18] Z. Zhang, M. Huang, Z. Zhao, F. Ji, H. Chen, and X. Zhu, "Memory-augmented dialogue management for task-oriented dialogue systems." *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 3, p. 34, 2019.

[19] J. E. Weston, "Dialog-based language learning." in *Advances in Neural Information Processing Systems*, 2016, pp. 829–837.

[20] R. Mitkov *et al.*, "Computer-aided generation of multiple-choice tests." in *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing*, 2003, pp. 17–22.

[21] M. Heilman and N. A. Smith, "Good question! statistical ranking for question generation." in *Proceedings of NAACL: Human Language Technologies*, 2010, pp. 609–617.

[22] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," *arXiv preprint arXiv:1705.00106*, 2017.

[23] S. Rao and H. Daumé III, "Answer-based adversarial training for generating clarification questions." *arXiv preprint arXiv:1904.02281*, 2019.

[24] X. Xue, J. Jeon, and W. B. Croft, "Retrieval models for question and answer archives." in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 475–482.

[25] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences." in *Advances in neural information processing systems*, 2014, pp. 2042–2050.

[26] D. Wang and E. Nyberg, "A long short-term memory model for answer sentence selection in question answering." in *Proceedings of ACL and IJNLP (Volume 2: Short Papers)*, 2015, pp. 707–712.

[27] Y. Tay, L. A. Tuan, and S. C. Hui, "Cross temporal recurrent networks for ranking question answer pairs." in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[28] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification." in *COLING*, 2016, pp. 3298–3307.
[29] Y. Wang, M. Huang, and L. Zhao, "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.
[30] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." *CoRR*, vol. abs/1803.01271, 2018.
[31] O. Wu, T. Yang, M. Li, and M. Li, "ρ-hot lexicon embedding-based two-level LSTM for sentiment analysis." *CoRR*, vol. abs/1803.07771, 2018.
[32] D. P.Kingma and J. L. Ba, "Adam: A method for stochastic optimization." *CoRR*, vol. abs/1412.6980, 2014.