Deep Human Answer Understanding for Natural Reverse QA

Rujing Yao[a,b], Linlin Hou[c], Lei Yang[a], Jie Gui[d], Ou Wu[a,*]

[a] *Center for Applied Mathematics, Tianjin University, China*
[b] *Department of Information Resources Management, Business School, Nankai University, China*
[c] *Center for Combinatorics, Nankai University, China*
[d] *Department of Computational Medicine and Bioinformatics, University of Michigan, USA*

Abstract

This study focuses on a reverse question answering (QA) procedure, in which machines proactively raise questions and humans supply the answers. This procedure exists in many real human-machine interaction applications. However, a crucial problem in human-machine interaction is answer understanding. Existing solutions have relied on mandatory option term selections to avoid automatic answer understanding. However, these solutions have led to unnatural human-computer interaction and negatively affected user experience. Thus, we propose a novel deep answer understanding network, AntNet, for reverse QA. The network consists of three new modules, namely, a skeleton attention for questions, a relevance-aware representation of answers, and a multi-hop-based fusion. Furthermore, to alleviate the negative influences of some quite difficult human answers, an improved self-paced learning strategy is proposed to train the AntNet by assigning different weights to training samples according to their learning difficulties. Given that 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 deep models. The effectiveness of the three modules

*Corresponding author

Email addresses: rjyao@tju.edu.cn (Rujing Yao), llhou@mail.nankai.edu.cn (Linlin Hou), yl7268@tju.edu.cn (Lei Yang), guijie@ustc.edu (Jie Gui), wuou@tju.edu.cn (Ou Wu)

and the improved self-paced learning strategy is also verified.

Keywords: Question answering (QA), Reverse QA, Answer understanding, Attention, Self-paced learning

1. Introduction

Automatic question answering (QA) is a crucial component in many human-machine interaction systems, such as intelligent customer service, because it can provide a natural means for humans to acquire information [1, 2]. In recent years, QA has received increasing attention in academic research and industry communities [3, 4, 5]. Questions are solely raised by humans, and answers are returned by machines in a conventional QA scenario, such as frequently asked questions (FAQ). The manner of selecting the best-matched answer is the key problem in this scenario [6].

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 the answers. This procedure is called reverse QA. Although this process has received minimal 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 nearly all 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 at least one of the fixed candidate items. This strategy is an engineering solution, in which the interaction between users and AI systems is unnatural.

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

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

[2] To our knowledge, only our early work [7] explored this issue. This study is an extension of our early work [7]. Nevertheless, a larger data corpus is compiled, and an entire new deep neural network is proposed.

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

This study proposes a new deep neural network, namely, the answer understanding network (AntNet), on the basis of the observations on a new data corpus and inspired by the related studies, such as aspect-based sentiment analysis [8, 9]. Furthermore, considering that some training samples are more difficult to train than others, an improved self-paced learning strategy which assigns different weights to different training samples during the training is proposed.

Given a machine-question and human-answer pair, AntNet extracts dense feature vectors for the question and the answer, and then fuses the two extracted vectors. A high-level dense feature vector is obtained and fed into a softmax layer for final answer understanding. Three new modules are included in AntNet. The first and second modules are the skeleton attention for questions and the relevance-aware representation of answers, respectively. The primary goal of the two modules is to exclude less important or disturbing information in questions and answers. The third module is the multi-hop-based fusion that is used to fuse answer and question vectors. The improved self-paced learning strategy is utilized in training to control the learning focus of easy and hard samples in training. Our proposed network is compared with existing methods and those modified from classical natural language processing deep models, such as Transformer [10]. The effectiveness of the improved SPL strategy is also verified.

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

[3]https://github.com/NlpResearchs/AntNet-ReverseQA

significantly outperforms the competing methods, and the improved SPL strategy further improves the performance of AntNet.

Our contributions are summarized as follows:

- A new problem, namely, human answer understanding for machine question, is initiated. A new corpus is also constructed for this new problem.
- A new deep neural network is proposed for the new problem. Our network contains standard modules widely used in existing related networks, such as the multi-hop attention module. New modules are also designed to better represent the input question and answer sentences, such as the skeleton attention and the relevance-aware answer representation modules.
- A new learning strategy is proposed to give priority to easy samples at first and then give priority to hard samples gradually during training.

2. Related Work

The most related study to reverse QA is question answering (QA). QA covers a wide range of tasks according to the application context. First, this section briefly reviews four related tasks, namely, text matching-based answer selection, multi-choice reading comprehension, question generation, and named entity recognition. Second, reverse QA is reviewed, and the differences between answer understanding in reverse QA and the three QA tasks are discussed.

2.1. Text matching-based answer selection

QA aims to return appropriate answers to users' questions. Therefore, the answers are usually selected from a corpus containing questions and answers on the basis of a text-matching model in many studies. In some studies, the model calculates the matching scores between the answers and questions in the corpus. The answers to questions with the highest matching score are then selected to return to users. Some other studies directly infer the matching score between the question and each candidate's answer.

In traditional QA methods, features of questions and answers are extracted using conventional methods, such as tf-idf [11], lexical cues [12], and

word order [13]. Thereafter, a similarity scoring function, such as cosine, is used to calculate the matching score.

In deep QA methods, the features of questions and answers are extracted using deep learning methods, such as convolutional neural network (CNN), LSTM, and Transformer [14, 15, 10]. An end-to-end framework is usually used to combine the deep feature extraction and successive matching function training [16, 17].

Inspired by the advantage of translation in modeling the relationship between words, Xue et al.[18] used a translation-based approach to solve the problem of mismatching. Subsequently, popular neural networks like CNN and LSTM were used in this task[19, 20]. Tay et al.[21] proposed a recurrent network using temporal gates to learn the interactions between question-answer pairs.

2.2. Multiple-choice reading comprehension

Multiple-choice reading comprehension (MCRC) aims to select the best answer from a set of options given a question and a passage. Unlike machine reading comprehension, in which the expected answer is directly contained in a given passage, answers in MCRC are non-extractive and may not appear in the original passage, thereby enabling rich types of questions, such as commonsense reasoning and passage summarization [22].

Numerous studies on MCRC model the relationship among the triplet of three sequences, namely, passage (P), question (Q) and answer (A), with a matching module to determine the answer. Zhu et al. [23] used hierarchical attention flow to explicitly model the option correlations, which are ignored in previous works. Zhang et al. [24] leveraged the bidirectional matching strategy to gather the correlation information among the triplet {P, Q, A}. The gated mechanism was then introduced to fuse the representations. In the matching process, Ran et al. [25] compared options at a word level to effectively collect option correlation information.

2.3. Question generation

Many QA algorithms require labeled QA pairs as training data. Although labeled data sets, such as the WikiQA dataset [26] for (text) QA have been proposed, these data sets are still with limited sizes because labeling is considerably expensive. This situation motivated the design of question generation

to generate natural language questions from information, in which the generated questions can be answered by the contents [27, 28]. In this manner, a large-scale QA corpus can be constructed.

Early research in question generation tackled question generation with a rule-based approach [29] or an overgenerate-and-rank approach [30], which relied heavily on well-designed rules or manually crafted features, respectively. To overcome these limitations, Du et al. [31] introduced a deep sequence-to-sequence learning approach to generate questions. Rao et al. [32] introduced generative adversarial networks (GANs) to generate questions that are significantly beneficial and specific to the context.

2.4. Named entity recognition

Named entity recognition (NER) is a fundamental task in natural language processing and is widely used in many applications, such as QA, text summarization, and machine translation [33, 34]. NER is mainly used to extract named entities from text, including persons, locations, and organizations. Recently, research on NER mainly has focused on low-resource, discontinuous and nested entities [35, 36, 37]. Ji et al. [38] proposed a novel bundling learning paradigm for the NER task, which does not need additional data annotations compared with multi-task learning. Liu et al. [39] proposed attention-informed mixed-language training, which achieves significant performance improvements with very few word pairs. Wang et al. [40] addressed discontinuous NER by finding the maximal cliques in the graph and connecting the spans in each clique. A two-stage entity identifier was proposed by Shen et al. [41] to address nested entity recognition.

The difference between NER and our answer understanding is mainly reflected in three aspects. First, their input data are different. In NER, only the concerned sentences are used as input, whereas in answer understanding, both the question and concerned answer sentences are used as input. Second, their corresponding classification problems are different. NER is a multi-class single-label classification problem, whereas answer understanding is a (non-standard) multi-class multi-label classification problem. Third, their main challenges are different. In answer understanding, irrelevant content and casually colloquial expressions are the main challenges, but low-resource, discontinuous and nested NER are the main challenges in the NER literature recently.

2.5. Reverse QA

Apart from meeting users' information requirements, machines in some real applications, such as telephone surveys and commercial intelligent customer service systems, are also required to proactively acquire the precise needs or feedback of users [42]. Accordingly, machines may choose to proactively raise questions to users and then analyze their answers. That is, machines are the questioners, and humans are the answerers. This process is a reverse of some text match-based QA processes (e.g., FAQ) and is called reverse QA in this study. Fig. 2 shows conventional FAQ and reverse QA processes.

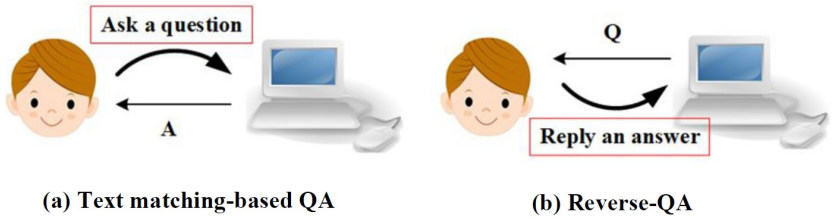

Figure 2: The difference between text match-based QA (e.g., FAQ) (a) and reverse QA (b).

The difference between answer understanding in reverse QA and text matching-based answer selection in QA is as follows. Text matching-based answer selection is a text retrieval approach, and the evaluation metrics (e.g., mean average precision (MAP) and mean reciprocal rank (MRR)) used for retrieval are usually applied. Consequently, the idea of learning to rank is typically adopted. Nevertheless, the answer understanding in reverse QA is transformed into an answer classification task. Fig. 3 shows the main difference between answer retrieval in text match-based answer selection in FAQ and answer classification in reverse QA[4].

The difference between answer understanding in reverse QA and the multi-choice reading comprehension (MCRC) in QA is as follows. First, from the viewpoint of classification, MCRC is a single-label classification task, whereas answer understanding (for multi-choice questions) in this study is a

[4]However, answer understanding investigated in this study is still a standard NLP task. Thus, text matching can also be utilized. Our preliminary experimental results show that a simple text matching module does not improve the performance of our proposed AntNet.

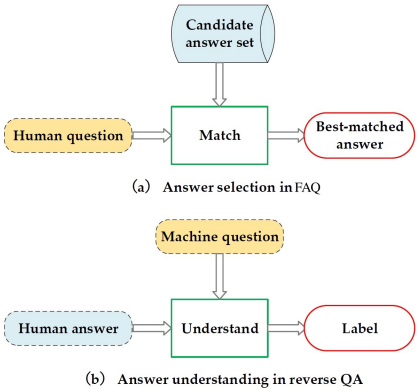

Figure 3: The main difference between answer selection in FAQ and answer understanding in reverse QA.

multi-label classification task[5]. Second, the option texts in MCRC are independent of other inputs (i.e., paragraphs and questions), whereas the option terms in this study are contained in the questions. Text matching is the key part of MCRC.

Question generation is apparently different from answer understanding investigated in this study. Nevertheless, inspired by question generation, answer generation will be explored in our future study to alleviate the labeling load.

3. Problem and Data

We first provide an analysis for answer understanding in reverse QA because it is rarely investigated.

3.1. Problem analysis

The primary difficulty in answer understanding results from the openness of the corresponding question. For example, the three machine questions (MQ) are as follows:

- MQ1: Do you like sports?

[5]Considering a machine question-human answer pair "Q: Which day can you come here, Monday, Tuesday, or Wednesday? A: Monday or Wednesday." The option terms "Monday" and "Wednesday" are the correct answers. In MCRC, only one answer is correct among the involved options.

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

MQ1 is a true/false (T/F) question, MQ2 is a multi-choice (MC) question, and MQ3 is nearly an open question. The difficulty of understanding the answers to these three questions is increasing. The answers for MQ3 are relatively difficult to understand considering the following answer examples: (1) "It depends on the weather," (2) "Competitive sports," and (3) "Water sports."

This study considers the T/F and MC questions. Consequently, answer understanding becomes a classification problem. The next subsection presents a formal description.

3.2. Problem formalization

As previously mentioned in Section 2.5, answer understanding for multi-choice (MC) questions can be categorized as a multi-label classification task. However, the number of categories for candidate labels for each question equals the number of option items contained in the question. Therefore, the number of categories for each question is likely to be different. The multi-label classification problem is usually transformed into one of the three existing problems, namely, binary classification, label ranking, and multi-class classification [43]. To tackle varied numbers of label categories, the strategy of the transformation to binary classification[6] is leveraged.

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," and "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. Each question can have arbitrary numbers of answers given by users. Let $s_{i,j}$ be the j-th answer for q_i. For MQ1, given an answer $s_{i,j}$, answer understanding aims 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

[6]Triple classification is actually used in this study.

an answer $s_{i,j}$, answer understanding equals three sub-classification tasks, i.e., the classification of $\{q_i, s_{i,j}, \textit{"swimming"}\}$, $\{q_i, s_{i,j}, \textit{"climbing"}\}$, and $\{q_i, s_{i,j}, \textit{"playing football"}\}$ 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. Thus, o in the triplet can be omitted in such question type.

3.3. Data construction

Existing QA and text classification benchmark data sets are inappropriate for training and evaluating reverse QA models. Thus, two data sets are compiled with a standard labeling process. The MC questions we studied are 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 30 graduate students, specifically 15 males and 15 females, are 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 31. 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, 1543 questions are obtained after deleting some invalid questions. The numbers of T/F and MC questions are 536 and 1007, respectively.

The questions are equally and randomly assigned to the 30 participants. Each question was given 18 to 25 answers. The participants also labeled the answers they generated considering that the other participants did not know what exactly the answer means. A new data corpus was obtained. Table 1 shows the details. The data corpus contains two data sets, namely, TData and MData. TData contains 536 T/F questions and 10,817 answers. Each question is associated with 20.18 answers on average. The average question length is 14.89, and the average answer length is 7.83. MData contains 1,007 MC questions and 23,445 answers. Each question is associated with 23.28

answers on average. The average question length is 22.82, and the average answer length is 7.84.

Table 1: Statistics of TData and MData. "Average Number" means the average number of answers corresponding to a question.

Data Set	Question Number	Answer Number	Average Number	Average Question Length	Average Answer Length
TData	536	10,817	20.18	14.89	7.83
MData	1007	23,445	23.28	22.82	7.84

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. Each sample consists of three components: question (i.e., q_i), answer (i.e., s_{ij}), and the associated label (l_{ij}) for them. The total number of samples is 10,817. For the MData, the number of option terms for each MC question is 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 3.2. Therefore, the same answer to the same question will have different labels for different option terms. Each sample consists of four components: question, option, answer, and label. The transformed MC training samples are 59,794. The type distribution of the samples is shown in Table 2.

Table 2: Type distribution of samples in TData and MData.

Data Set	Samples			
	Total	Affirmative	Negative	Uncertain
TData	10,817	4,610	4,452	1,755
MData	59,794	20,929	28,876	9,989

Some illustrative examples for T/F and MC questions are showed in Table 3. In TData, the categories of answers A1, A3, and A5 are easy to judge. However, the machine has difficulty understanding the true meaning of answers A2, A4, and A6. For example, answer A2 is a true answer to Q1, but can easily be identified as a false answer. The MC questions are the same. For the A10 answer to Q5 in MData, the human does not directly answer what kind of ball game he likes but answers who his favorite football star is.

The human means he likes football, which is hard for a machine to judge. Thus, understanding users' answers is not a trivial task. The main challenge for the classification investigated in the study is that the representation of human answers is irregular. Furthermore, human answers often contain slang or colloquial words, which heavily increase the difficulty of the task.

Table 3: Some illustrative examples for T/F and MC questions.

Dataset	Question	Human answer	Label
TData	Q1:今晚的音乐剧你觉得好不好? (Do you think tonight's musical is good?)	A1:这是我看过的最棒的音乐剧 (This is the best musical I've ever seen.)	True
		A2:和这个比，其他的都是垃圾 (Compared with this, everything else is rubbish.)	True
	Q2：您觉得就业压力大吗? (Do you think the employment pressure is high or not?)	A3:一点也不大 (Not at all.)	False
		A4:头发都掉完了，你说呢 (The hair is all gone. What do you think?)	True
	Q3：运动前会做准备运动吗? (Do you warm up before exercise?)	A5:运动前必备操作 (This is a must before exercise.)	True
		A6：不做准备运动不怕拉伤吗 (Aren't you afraid of strain if you don't warm up?)	True
MData	Q4: 面对新奇的运动方式你会尝试了解还是默默离开? (In the face of novel sports, will you try to understand or leave silently?)	A7: 应该会默默离开 (I should leave silently.)	"True" for "leave silently" and "False" for "try to understand"
		A8: 别人尝试过我再尝试 (I will try what others have tried.)	"True" for "leave silently" and "False" for "try to understand"
	Q5: 羽毛球、乒乓球、足球您更喜欢哪个? (Which do you prefer: badminton, table tennis or football?)	A9: 羽毛球 (Badminton.)	"True" for "badminton"; "False" for both "table tennis" and "football"
		A10:我想要向梅西学习 (I want to learn from Messi.)	"True" for "football"; "False" for both "badminton" and "table tennis"
	Q6: 您希望在大城市工作还是小城市? (Would you like to work in a big city or a small city?)	A11: 大城市 (A big city.)	"True" for "a big city" and "False" for "a small city"
		A12:这年头还有人愿意去小城市? (Is anyone still willing to go to a small city these days?)	"True" for "a big city" and "False" for "a small city"

4. Methodology

Section 3.2 describes that answer understanding is transformed into an answer classification problem. The first step is obtaining the deep representations of the machine-question and human-answer pair and a given option term. In addition, questions provide the context for answer understanding. The final dense representation should consider the contextual dependency between questions and answers.

The related research in text classification and aspect-based sentiment analysis inspired us to propose a new deep model called AntNet. Fig. 4 shows the main structure of this model.

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

The AntNet input is the triplet $\{q_i, s_{i,j}, o_{i,k}\}$, where $o_{i,k}$ is indicated by an indicator vector. The indicator is set to a zero vector for all samples in T/F questions, and the option indicator is set to a one-hot vector in MC questions. The left part of AntNet deals with the input of q_i and $o_{i,k}$ to generate two representations. The first representation characterizes the

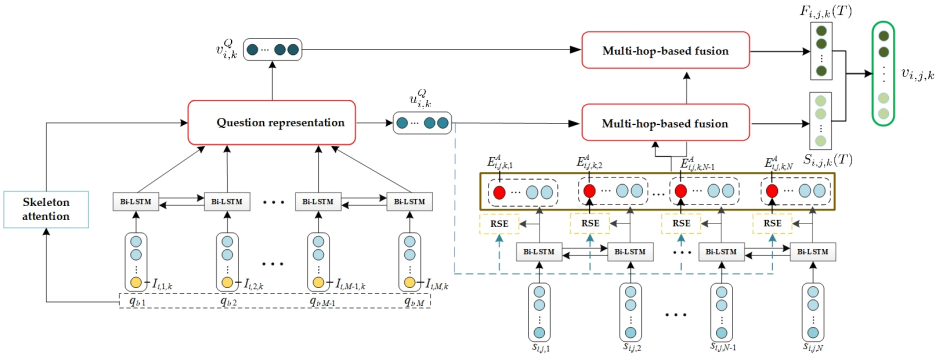

Figure 4: The structure of AntNet.

combination of q_i and $o_{i,k}$[7], whereas the second representation characterizes important information, which is called skeleton (Chinese) words for questions in this study. The first and second representations are called full and skeleton representations, respectively.

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 substantially characterize the relevance cues contained in the answers, which consider 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 and the improved SPL strategy proposed by us.

4.1. Skeleton attention

Question texts usually contain redundant[8] or even disturbed words, which may negatively influence answer understanding. The skeleton information in a question should be extracted. Skeleton information refers to words that directly affect how users respond.

Skeleton information extraction can be performed in a supervised manner. Alternatively, skeleton words are manually labeled for a set of training

[7] $o_{i,k}$ is indicated by $\{I_{i,1,k}, \cdots, I_{i,M,k}\}$ in Fig. 4.2, which will be mentioned in detail in Section 4.2.

[8] These words may be used for enhancing the interestingness of the interaction.

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, it is difficult to provide an explicit and formal definition for skeleton words, thus making human labeling difficult. Therefore, this study proposes an attention-based manner.

In this study, a training sample is a triplet $\{q_i, s_{i,j}, o_{i,k}\}$, where q_i is the i-th question, $s_{i,j}$ is the j-th answer for q_i, 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 the fact that many training samples in this study share the same element "q_i". That is, each question (q_i) corresponds to multiple answers $(s_{i,1}, \cdots, s_{i,j}, \cdots, s_{i,J})$, leading to 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 of $s_{i,j}$. An attention score can be calculated for $q_{i,m}$. The calculation pipeline is shown in Fig. 5, and the calculation is described as follows:

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

where $sim(q_{i,m}, s_{i,j,n}) = \tilde{q}_{i,m}^T W_s \tilde{s}_{i,j,n}$ calculates the similarity of two words according to their word embeddings, $\tilde{q}_{i,m}$ represents the embedding vector of the word $q_{i,m}$, $\tilde{s}_{i,j,n}$ represents the embedding vector of the word $s_{i,j,n}$, the matrix W_s are learned during training, and J is the number of answers to i-th question.

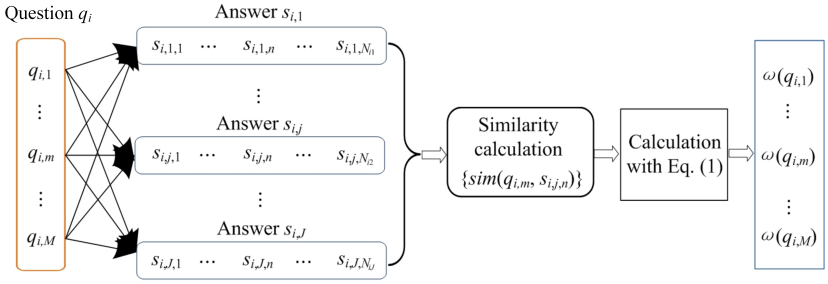

Figure 5: Skeleton attention score calculation pipeline.

The scores calculated by Eq. 1 are then normalized as attention scores. Fig. 6 shows several questions and their associated attention scores on each Chinese word calculated using Eq. 1. The words with higher scores are key words in their corresponding questions. The scores of such words as "您 (you)," "是 (is)," and "还是 (or)" are low in most sentences. The words that are directly related to user options, such as "跑步 (run)," "兴趣 (interest)," "质量 (quality)," and "在家 (at home)" have high scores. Fig. 7 shows a question with associated attention scores for four different answers on each Chinese word. In the attention scores of the question corresponding to the first answer, "更 (more)," "看重 (value)," and "质量 (quality)" are closely related to the answer, resulting in a high attention score. Similarly, in the attention scores of the question corresponding to the second and fourth answer, "价格 (price)" and "质量 (quality)" have high scores. In the attention scores of the question corresponding to the third answer, the word "价格 (price)" has the highest attention score. No explicit contents are related to the question in the third answer. Thus, the scores for each word are not much different. Nonetheless, the attention scores for "价格 (price)" and "质量 (quality)" are ranked first and second, respectively.

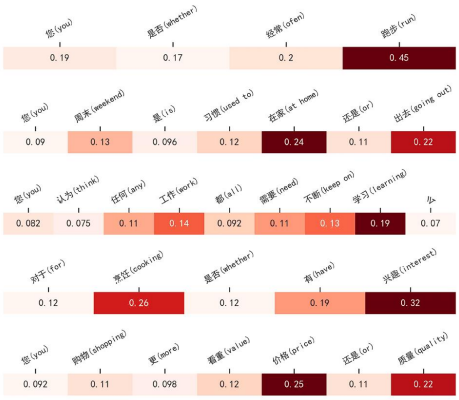

Figure 6: Attention scores for words in five questions. All the data in this study are in Chinese. To facilitate English readers, the Chinese words in the above questions are translated into English.

4.2. Question representation

AntNet considers two-level representations. The first-level representation (i.e., skeleton representation) characterizes the skeleton information in the

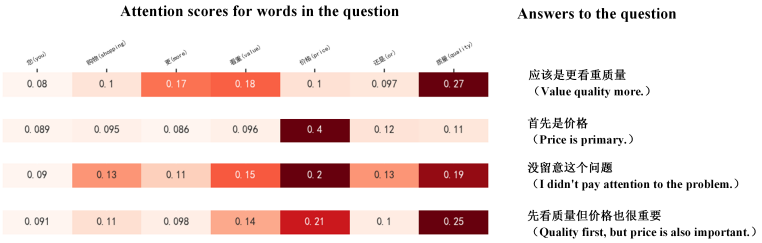

Figure 7: Attention scores for words in a question for four different answers. All the data in this study are in Chinese. To facilitate English readers, the Chinese words in the above question and answers are translated into English.

question, whereas the second-level representation (i.e., full representation) characterizes the entire 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 $o_{i,k}$[9]. If $q_{i,m}$ is in $o_{i,k}$, then $I_{i,m,k} = 1$; otherwise $I_{i,m,k} = 0$.

After the encoding of BiLSTM on q_i, the hidden representation of each word of q_i (given $o_{i,k}$) is defined as follows:

$$h^Q_{i,m,k} = BiLSTM_Q(h^Q_{i,m-1,k}, h^Q_{i,m+1,k}, \tilde{q}_{i,m}, I_{i,m,k}), \tag{2}$$

where $h^Q_{i,m,k} \in \mathbb{R}^d$.

Given that the skeleton attention is calculated using Eq. 1, the skeleton representation for a question q_i (given $o_{i,k}$) is calculated as follows:

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

where $\omega(q_{i,m})$ is the skeleton attention for the m-th word of the i-th question, and $\omega(q_{i,m})$ is calculated by Eq. 1.

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

[9]Some option terms are phrases.

The calculation is described as follows:

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

where $w_a \in \mathbb{R}^d$, $a^Q_{i,m,k}, att^Q_{i,m,k} \in \mathbb{R}$, $v^Q_{i,k} \in \mathbb{R}^d$. w_a is a learnable vector and it can be viewed as the query vector in the attention calculation. $h^Q_{i,m,k}$ is the key as well as the value vector simultaneously.

4.3. Relevance-aware answer representation

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

$$h^A_{i,j,n} = BiLSTM_A(h^A_{i,j,n-1}, h^A_{i,j,n+1}, \tilde{s}_{i,j,n}), \tag{5}$$

where $h^A_{i,j,n}$ ($\in \mathbb{R}^d$) is the hidden vector of the n-th word of the j-th answer to the i-th question, and $\tilde{s}_{i,j,n}$ represents the embedding vector of the word $s_{i,j,n}$. To maintain the naturalness of the entire 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. Thus, a score (denoted as $p^A_{i,j,k,n}$) is calculated to measure the relevance between each word in the answer texts (i.e., $s_{i,j,n}$) and each option term in the question (i.e., $o_{i,k}$) using the following equation:

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

where $u^Q_{i,k}$ is the skeleton representation of a question q_i for a option term $o_{i,k}$; W_p and b_p are learnable parameters.

The length of $p^A_{i,j,k,n}$ is substantially smaller than $h^A_{i,j,n}$ in our practical implementation. Consequently, the proportion of the $p^A_{i,j,k,n}$ part is relatively small in the concatenated vectors, thereby limiting the advantages of the relevance vectors. We adopt the trick used in [44] in our implementation. The length of $p^A_{i,j,k,n}$ is enlarged as follows:

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

where $1_{N_e \times 1}$ is an N_e-dimensional vector, $E^A_{i,j,k,n}$ is the enlarged vector, and parameter N_e is used to increase the length of $p^A_{i,j,k,n}$. Fig. 8 shows the steps of the relevance score calculation and dimensionality enlarging. 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 as follows:

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

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

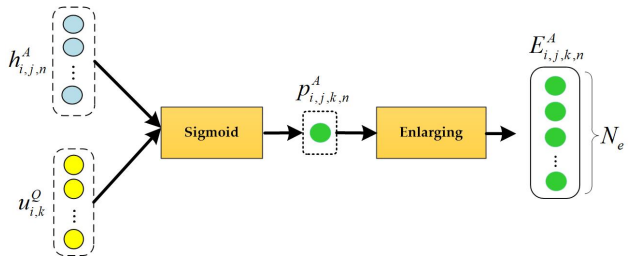

Figure 8: The relevance score calculation and dimension enlarging.

4.4. Multi-hop-based fusion

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

Inspired by ABSA [45], a multi-hop-based question-answer fusion module is introduced. This module can substantially 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 separately input into the multi-hop-based fusion module. Fig. 9 shows the multi-hop-based fusion. The left part and the right part are the iterative approaches for $v^Q_{i,k}$ and $u^Q_{i,k}$, respectively, given $h'^A_{i,j,k}$.

The calculation with $v^Q_{i,k}$ and $h'^A_{i,j,k}$ is used 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. 9) is

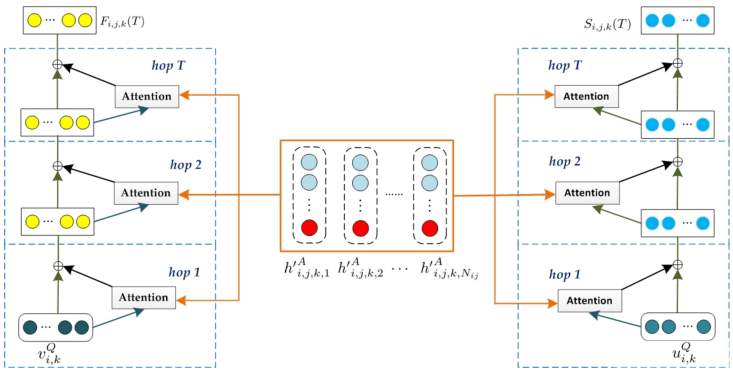

Figure 9: Multi-hop-based fusion for the involved question (two feature vectors $u^Q_{i,k}$ and $v^Q_{i,k}$) and answer ($h'^A_{i,j,k}$).

computed as follows:

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

where $W^{(1)}_m$, $W^{(1)}_h$, $W^{(1)}_x$, and $b^{(1)}$ are learnable parameters.

An active module is used to obtain the following new vector:

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

where W_{f1}, W_{f2}, and b_f are learnable parameters. $F_{i,j,k}(1)$ is also the input of the second hop (hop 2 in Fig. 9).

The preceding step is iterated T times to obtain the feature vector $F_{i,j,k}(T)$. Lastly, $F_{i,j,k}(T)$ from the full representation $v^Q_{i,k}$ and $S_{i,j,k}(T)$ from the skeleton question representation $u^Q_{i,k}$ are concatenated into one representation vector:

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

The predicted label is calculated as follows:

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

where W and b are learnable parameters.

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

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

4.5. The improved self-paced learning strategy

As described in Section 3.3, irrelevant answers and answers containing slang or colloquial words heavily increase the difficulty of the model training. To alleviate the negative influences of some quite difficult answers, a self-paced learning strategy is utilized. Self-paced learning (SPL) [46] is motivated by the human learning procedure that preliminary knowledge is studied in the junior stage, and sophisticated knowledge is studied in the senior stage. In SPL, easy training samples (i.e., samples with low training losses) are assigned with high weights, and hard samples (i.e., samples with high training losses) are with low weights in the early training stage. The weights of hard samples are gradually increased to be equal to easy samples with the increase of the training epoch. Let $v_i \in \{0, 1\}$ be the weight of the training sample x_i. In the k-th epoch, the optimization with SPL is as follows:

$$\begin{aligned} &\min_{\Theta} \ \sum_i v_i l(f(x_i, \Theta), y_i) \\ &\text{s.t.} \quad v_i = \begin{cases} 1 & \text{if } l\left(f\left(x_i, \Theta\right), y_i\right) < \frac{1}{\mathcal{K}} \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{14}$$

where $f\left(x_i, \Theta\right)$ is the model parameterized by Θ, $l\left(f\left(x_i, \Theta\right), y_i\right)$ is the loss of the i-th sample, and $\mathcal{K}$ is a hyper-parameter. $\mathcal{K}$ determines the number of samples to be considered, and the value of $\mathcal{K}$ is iteratively reduced. The objective function indicates that the weights of samples are set to 0 when losses are larger than $\frac{1}{\mathcal{K}}$, and more samples will participate in the model training when the value of $\frac{1}{\mathcal{K}}$ is increased.

SPL is an easy-first weighting strategy. Contrarily, in some other sample weighting strategies, such as Focal loss [47], hard samples are assigned (relatively) high weights. The effectiveness of these hard-first weighting methods

is also verified on numerous learning tasks. Motivated by these hard-first methods, we propose an improved SPL weighting strategy. In our improved SPL, easy training samples (i.e., samples with low training losses) are still assigned with high weights, and hard samples (i.e., samples with high training losses) are still with low weights in the early training stage. Nevertheless, the weights of hard samples are gradually increased to be larger than easy samples with the increase of the training epoch. Alternatively, easy-first weighting is leveraged in the early training stage, and hard-first weighting is leveraged in the late training stage. In the k-th epoch, the optimization with our improved SPL strategy is as follows:

$$\begin{aligned} \min_{\Theta} \ & \sum_i v_i l(f(x_i, \Theta), y_i) \\ \text{s.t.} \quad & v_i = \frac{1}{1+e^{-\alpha(l_i-\tau)sign(k-k_0)}} \ , \\ & \tau = \frac{k+sign(k_0-k)k_0}{k_0}\tau_0 \end{aligned} \tag{15}$$

where α, k_0, and τ_0 are hyper-parameters; $l_i = l(f(x_i, \Theta), y_i)$. When $k \leq k_0$, the weights of easy examples are larger than those of hard samples. When $k > k_0$, the weights of easy examples are smaller than those of hard samples.

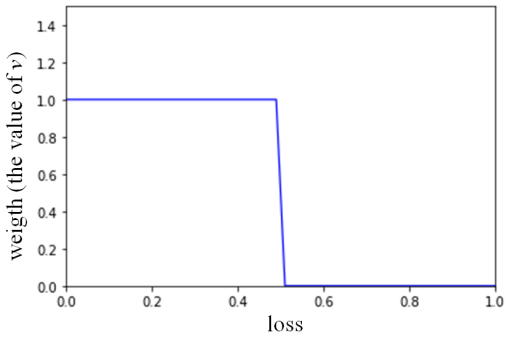

Figure 10: The curve of SPL when the threshold is 0.5.

We plot the weight curves of SPL and the improved SPL to show the difference. Fig. 10 shows the curve of SPL when the loss locates in [0, 1]. Fig. 11 shows the curves of the improved SPL. Easy samples have higher weights than hard ones before the 10th epoch, whereas their weights are smaller than hard ones after the 10th epoch.

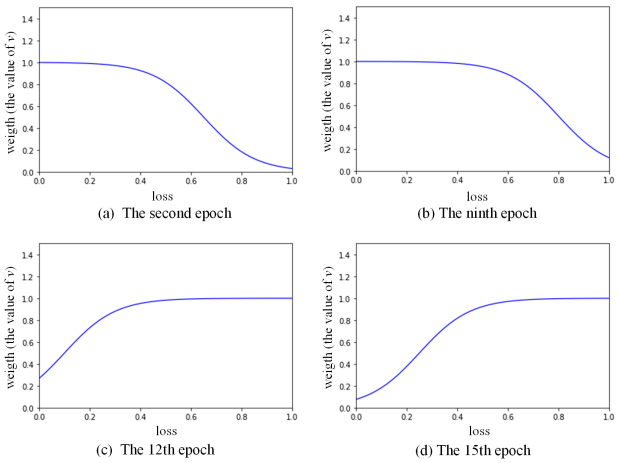

Figure 11: The curves of the improved SPL when α is set to 10, k_0 is set to 10, and τ_0 is set to 0.5.

5. Experiments

This section presents the evaluation of the proposed AntNet and the improved SPL strategy.

5.1. Evaluation of AntNet

For AntNet, the entire network and the three key modules, namely, skeleton attention for questions, relevance-aware representation of answers, and multi-hop-based fusion, are evaluated.

5.1.1. 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 the question and answer texts.
- **RAM** [8]: RAM leverages the hidden vectors of BiLSTM as memory vectors. Then, GRU is used to construct a multi-hop-based fusion for memory and input target vectors. The final dense vector contains

information from sentences and targets. This study takes question texts as target texts.

- **ATAE** [48]: 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)**: Questions and answers are concatenated and input into the standard transformer.

- **Semi-IAN** [7]: Semi-interactive attention network (Semi-IAN) is our early proposed network related to answer understanding in reverse QA. The interaction between questions and answers are modeled. Semi-IAN is based on an ABSA network called interactive attention network (IAN) [45].

- **(Python) Regularized Matching (RM)**: This method is an engineering solution that matches pre-defined key words or phrases or their combinations.

Our proposed method consists of several new modules. To investigate the validity of three major components, namely, skeleton attention, 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**: The entire AntNet with all introduced key components.

- **AntNet-SA**: The AntNet without the skeleton attention.

- **AntNet-RR**: The AntNet without the relevance-aware representation.

- **AntNet-MF**: The AntNet without the multi-hop-based fusion.

- **AntNet-SA-RR**: The AntNet without the skeleton attention and the relevance-aware representation.

- **AntNet-RR-MF**: The AntNet without the relevance-attention representation and the multi-hop-based fusion.
- **AntNet-MF-SA**: The AntNet without the multi-hop-based fusion and the skeleton attention.

Given that the answer understanding for reverse QA is investigated from a classification perspective, the classification accuracy and F1 score are used as the performance metrics.

5.1.2. Implement details

Two data sets, 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 a 4:1 proportion. Four folds are used for training, and the remainder is used for testing.

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

For all deep learning methods, the lengths of questions and answers are truncated to 33. The dropout rate is set to 0.2. We use 256-dimension Word2Vector embeddings trained on our own corpus. Out-of-vocabulary words are randomly initialized with word embeddings. The dimension of the word vector pre-trained by BERT is 768. We minimize the loss function using the Adam optimizer [49].

For BiLSTM and ATAE, the epochs, batch size, learning rate, and the dimension of hidden vectors are set to 32, 32, 5e-4, and 300, respectively. For RAM, the epochs, batch size, learning rate, and the dimension of hidden vectors are set to 32, 16, 5e-4, and 128, respectively. The number of hops is set to three. For Transformer, the epochs, batch size, learning rate, and the dimension of hidden vectors are set to 24, 16, 5e-3, and 128, respectively. For Semi-IAN, the epochs, batch size, learning rate, and the dimension of hidden vectors are set to 32, 16, 5e-4, and 128, respectively. In ρ-hot encoding, the size k is searched in {1, 2, 4, ..., 16}; the parameter ρ is searched in {0.1, 0.2, ..., 1}. For AntNet, the epochs, batch size, learning rate, and the dimension of hidden vectors are set to 32, 16, 5e-4, and 128, respectively. The number of hops is set to three and five in TData and MData, respectively. All the mentioned models are trained with Tensorflow.

5.1.3. Evaluation of the entire AntNet network

Table 4 presents the main results (classification accuracies and F1 scores) of the competing methods on the two data sets. AntNet achieves the highest

accuracies on both data sets. Compared with the state-of-the-art network, Transformer, the results are significantly improved. The relatively poor performance of Transformer may result from the small training size.

Table 4: Experiments on TData and MData.

Method	TData		MData	
	Accuracy	F1	Accuracy	F1
BiLSTM (A)	0.7375	0.7196	0.6701	0.6681
BiLSTM (Q+A)	0.7196	0.6982	0.6738	0.6868
RAM	0.7503	0.7435	0.7036	0.6860
ATAE	0.7458	0.7361	0.7064	0.7446
Transformer (A)	0.7435	0.7343	0.6741	0.6525
Transformer (Q+A)	0.7167	0.6911	0.6966	0.6537
Semi-IAN	0.7485	0.7427	0.7086	0.6871
AntNet	**0.7986**	**0.7923**	**0.8419**	**0.8517**

The existing answer understanding method (i.e., Semi-IAN) is inferior to RAM. Semi-IAN is a slight variation of the ABSA network IAN. Given that RAM is also an ABSA method, RAM unsurprisingly outperforms Semi-IAN. Among these methods, the RM method has the lowest accuracy of 50.38% on average. Therefore, a machine learning-based approach is essential.

Table 5: Experiments on TData with DuReader yes/no Data.

Method	Accuracy	F1
BiLSTM (A)	0.7445	0.7334
BiLSTM (Q+A)	0.7454	0.7423
RAM	0.7701	0.7674
ATAE	0.7492	0.7201
Transformer (A)	0.7529	0.7436
Transformer (Q+A)	0.7267	0.7126
Semi-IAN	0.7523	0.7485
AntNet	**0.8045**	**0.8048**

An existing QA corpus DuReader [50] is used to pre-train the involved

Table 6: Experiments on TData and MData with BERT.

Method	TData		MData	
	Accuracy	F1	Accuracy	F1
BiLSTM (A)	0.7492	0.7369	0.6864	0.6721
BiLSTM (Q+A)	0.7511	0.7442	0.6857	0.6932
RAM	0.7723	0.7693	0.7237	0.6981
ATAE	0.7521	0.7233	0.7103	0.7557
Transformer (A)	0.7601	0.7529	0.6863	0.6704
Transformer (Q+A)	0.7355	0.7221	0.7069	0.6839
Semi-IAN	0.7607	0.7563	0.7118	0.6992
AntNet	**0.8136**	**0.8067**	**0.8614**	**0.8705**

models. DuReader[10] is a large-scale real-world Chinese dataset with three types of questions, namely, "description," "entity," and "yes/no". The "yes/no" samples contain the information required for T/F questions, but the option term is missing for MC questions. Thus, the corpus can only be used for pre-training for T/F questions. Table 5 shows the results. The performances of all the competing methods are improved, although the increase is not significant.

Furthermore, a pre-trained model BERT [51] is also used in our experiments, and the experimental results are shown in Table 6. The performances of all the competing methods are improved compared to those in Table 4.

5.1.4. Evaluation of the different modules of AntNet

This subsection verifies 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-SA, AntNet-RR, AntNet-MF, AntNet-SA-RR, AntNet-SA-MF, AntNet-RR-MF, and the entire network AntNet.

Table 7 shows the competing results on the two data sets: TData and MData. All variations without a certain type of key module achieve inferior accuracies compared with the full version of AntNet. The performances of the variations without two key modules decrease heavily. These comparisons indicate that the three key modules are beneficial in answer understanding.

The comparison of the three variations shows that AntNet-SA-MF (AntNet

[10]http://ai.baidu.com/broad/download?dataset=dureader

Table 7: Results of AntNet and its variations (without certain key modules) on TData and MData.

Method	TData		MData	
	Accuracy	F1	Accuracy	F1
AntNet	**0.8045**	**0.8048**	**0.8419**	**0.8517**
-SA	0.7863	0.7774	0.8238	0.8374
-RR	0.7907	0.7790	0.7107	0.7094
-MF	0.7740	0.7551	0.7050	0.6897
-SA-RR	0.7760	0.7681	0.7033	0.7026
-SA-MF	0.7686	0.7494	0.6809	0.6736
-RR-MF	0.7714	0.7523	0.6943	0.6915

without SA and MF) obtains the lowest accuracies. On MData, the accuracy achieved by AntNet-SA-MF is approximately 16.1% lower than that by AntNet.

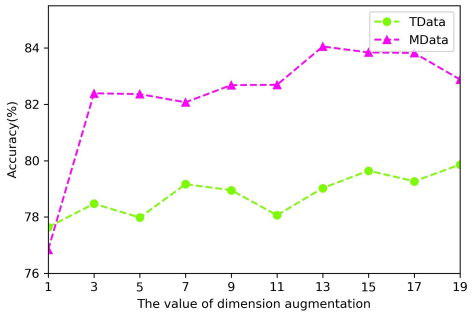

Figure 12: Understanding accuracies under the different values of dimension augmentation (i.e., N_e) for relevance-aware representation.

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 AntNet under different augment parameters N_e in Eq. (7). Fig. 12 shows the accuracies of 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 values equal 13 and 19, AntNet achieves the maximum accuracies on both data sets.

In the multi-hop module, the number of hops is also an important parameter. We perform experiments to explore the relationship between hop count

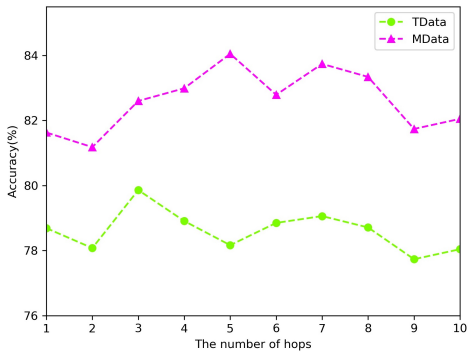

Figure 13: Accuracies under the different numbers of hops for multi-hop-based fusion.

and final accuracy. Fig. 13 shows the accuracies of AntNet under different numbers of hops.

The number of hops also influences the final performance. The highest value (when the number equals three) is nearly 2% higher than the lowest value (when the number equals nine) on TData. On MData, the overall trend increases, and the accuracy is the highest when the number equals five.

5.2. Evaluation of the improved SPL

To evaluate the performance of the improved SPL strategy, we compare SPL and the improved SPL on TData and MData. Word2Vector and BERT are adopted, and the results are presented in Table 8.

For SPL, the initial weight of $\mathcal{K}$ is searched in {5, 10, 15}, and is reduced by a factor 1.2 at each iteration. For the improved SPL, α is searched in {5, 10}, τ_0 is searched in {0.5, 1.5}, and k_0 is set to 10. Other parameter settings are the same as those in Section 5.1.2.

The experimental results show that the improved SPL strategy is effective. Compared with AntNet with SPL, AntNet with the improved SPL also shows significant improvement.

For AntNet pre-trained on DuReader, SPL and the improved SPL are also compared. The results in Table 9 show that the improved SPL strategy is effective.

In the improved SPL, α, τ_0, and k_0 are important parameters. We perform experiments to explore the relationship between α and accuracy, the relationship between τ_0 and accuracy, and the relationship between k_0 and accuracy. Figs. 14, 15, and 16 show the accuracies of AntNet under different

Table 8: Results of AntNet with SPL and the improved SPL on TData and MData

Embedding	Method	TData		MData	
		Accuracy	F1	Accuracy	F1
Word2Vector	AntNet	0.7986	0.7923	0.8419	0.8517
	AntNet with SPL	0.8133	0.7997	0.8492	0.8556
	AntNet with the improved SPL	**0.8249**	**0.8154**	**0.8641**	**0.8733**
BERT	AntNet	0.8136	0.8067	0.8614	0.8705
	AntNet with SPL	0.8226	0.8146	0.8707	0.8796
	AntNet with the improved SPL	**0.8393**	**0.8270**	**0.8832**	**0.8901**

Table 9: Results of AntNet with SPL and the improved SPL on TData with DuReader.

Method	Accuracy	F1
AntNet	0.8045	0.8048
AntNet with SPL	0.8177	0.8089
AntNet with the improved SPL	**0.8298**	**0.8196**

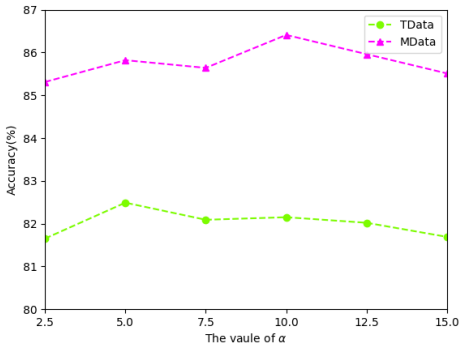

Figure 14: Accuracies under the different value of α.

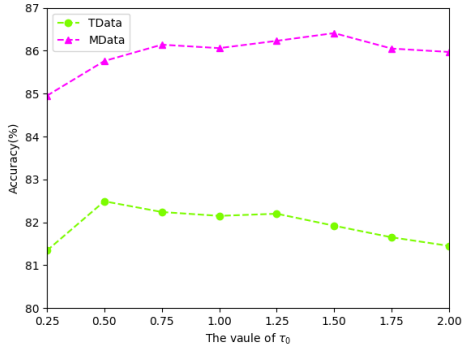

Figure 15: Accuracies under the different value of τ_0.

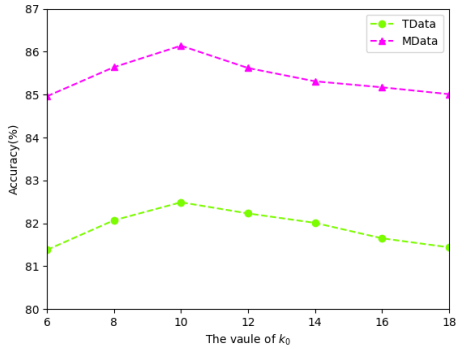

Figure 16: Accuracies under the different value of k_0.

values of α, τ_0, and k_0.

The values of α, τ_0, and k_0 all influence the final performance. For TData and MData, when α is greater than 5 and 10, the accuracies gradually decrease, respectively; when τ_0 is greater than 0.5 and 1.5, the accuracies gradually decrease, respectively; when k_0 equals 10, AntNet with the improved SPL achieves the best results on both data sets.

5.3. Discussion

We empirically analyze the error understanding answers in the test set to scrutinize the performance of AntNet substantially. The results show that errors are prone to occur for answers containing implicit preference information. In particular, once the implicit information contains negative or positive words, they are likely to be error judged. Table 10 shows several examples of answers containing implicit information. The first question belongs to the MC type. Thus, each label should correspond to an option term such as "skirts" and "pants."

Table 10: Examples in which human answers contain implicit relevance hints.

Machine question	**Human answer**	**Label**
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"

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

Attention is the core of deep neural networks in NLP [10]. The following example is visualized to facilitate the analysis of the effectiveness of the multi-hop attention used in this study.

In hop1 shown in Fig. 17, the attention score for the Chinese word "怎么(how)" is small. Nevertheless, in hop4, its attention score becomes high,

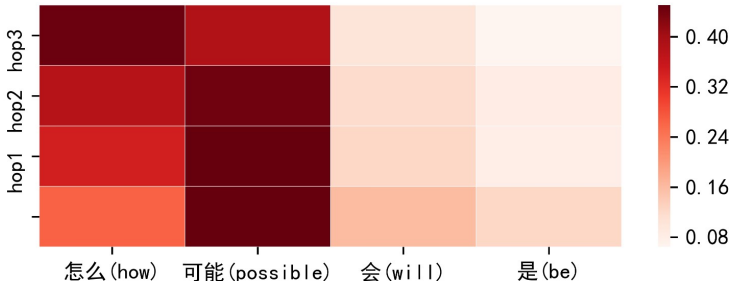

Figure 17: Multi-hop attention scores for an answer sentence.

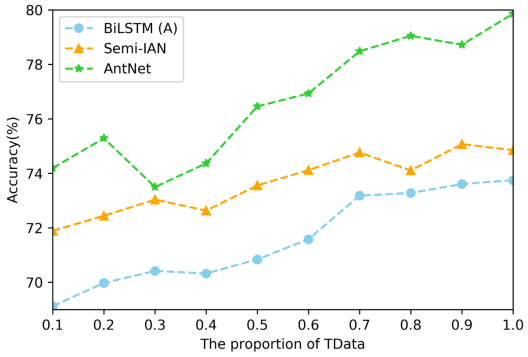

Figure 18: Accuracies under the different proportions of training data on TData.

which is reasonable because the Chinese word is quite important for answer understanding.

We also investigate the relationship between training data and model performances. Fig. 18 shows the variations of performances under different proportions of training data on TData. With the increase of training data, the performances of the three methods, AntNet, BiLSTM, and semiIAN, also increased. Nevertheless, when the training data is small, the performance of AntNet is also relatively good. Similar observations are obtained on MData.

6. Conclusion

The automatic understanding of human answers in reverse QA can make interactions natural and improve user experiences. However, this topic receives little attention in the 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 to understand the answers for the two most common types of questions. We also propose an improved self-paced learning strategy to improve the performance of AntNet further.

AntNet utilizes two types of questions and a relevance-aware presentation for answer texts. The multi-hop-based fusion module is used to model the contextual dependency between questions and answers. The improved SPL method combines the hard-first and the easy-first weighting strategies. The experimental results indicate that AntNet is significantly better than the existing method and state-of-the-art NLP models with direct variations. The improved SPL improves the performance of AntNet.

7. Acknowledgments

We thank Dr. Guan Luo, Dr. Xiaodong Zhu, and Prof. Qinghua Hu for their contributions to data collection and our early proposed model, SemiIAN, in the PAKDD paper. We also appreciate the anonymous reviewers for their insightful comments and constructive suggestions.

References

[1] B. Hixon, P. Clark, H. Hajishirzi, Learning knowledge graphs for question answering through conversational dialog, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for

Computational Linguistics: Human Language Technologies, 2015, pp. 851–861.

[2] P. Wang, L. Ji, J. Yan, D. Dou, N. D. Silva, Y. Zhang, L. Jin, Concept and attention-based cnn for question retrieval in multi-view learning, ACM Transactions on Intelligent Systems and Technology (TIST) 9 (4) (2018) 1–24.

[3] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, R. Socher, Ask me anything: Dynamic memory networks for natural language processing, in: International conference on machine learning, 2016, pp. 1378–1387.

[4] D. Thukral, A. Pandey, R. Gupta, V. Goyal, T. Chakraborty, Diffque: Estimating relative difficulty of questions in community question answering services, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (4) (2019) 1–27.

[5] X. Zhan, Y. Huang, X. Dong, Q. Cao, X. Liang, Pathreasoner: Explainable reasoning paths for commonsense question answering, Knowledge-Based Systems 235 (2022) 107612.

[6] C. Xiong, V. Zhong, R. Socher, Dynamic coattention networks for question answering, arXiv preprint arXiv:1611.01604 (2016).

[7] Q. Yin, G. Luo, X. Zhu, Q. Hu, O. Wu, Semi-interactive attention network for answer understanding in reverse-qa, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, pp. 3–15.

[8] 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.

[9] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm., in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018, pp. 5876–5883.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[11] G. Salton, C. T. Yu, On the construction of effective vocabularies for information retrieval, Acm Sigplan Notices 10 (1) (1973) 48–60.

[12] K. Khalifa, N. Omar, A hybrid method using lexicon-based approach and naive bayes classifier for arabic opinion question answering., J. Comput. Sci. 10 (10) (2014) 1961–1968.

[13] E. Hovy, U. Hermjakob, C.-Y. Lin, The use of external knowledge in factoid qa, in: TREC, Vol. 2001, 2001, pp. 644–652.

[14] M. Tan, C. Dos Santos, B. Xiang, B. Zhou, Improved representation learning for question answer matching, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 464–473.

[15] Y. Tay, M. C. Phan, L. A. Tuan, S. C. Hui, Learning to rank question answer pairs with holographic dual lstm architecture, in: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, 2017, pp. 695–704.

[16] Z. Wang, W. Hamza, R. Florian, Bilateral multi-perspective matching for natural language sentences, arXiv preprint arXiv:1702.03814 (2017).

[17] 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, arXiv preprint arXiv:1804.09541 (2018).

[18] X. Xue, J. Jeon, 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.

[19] B. Hu, Z. Lu, H. Li, Q. Chen, Convolutional neural network architectures for matching natural language sentences, Advances in neural information processing systems 27 (2014) 2042–2050.

[20] D. Wang, E. Nyberg, A long short-term memory model for answer sentence selection in question answering, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), 2015, pp. 707–712.

[21] Y. Tay, L. A. Tuan, S. C. Hui, Cross temporal recurrent networks for ranking question answer pairs., in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018, pp. 5512–5519.

[22] G. Lai, Q. Xie, H. Liu, Y. Yang, E. Hovy, Race: Large-scale reading comprehension dataset from examinations, arXiv preprint arXiv:1704.04683 (2017).

[23] H. Zhu, F. Wei, B. Qin, T. Liu, Hierarchical attention flow for multiple-choice reading comprehension, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018, pp. 6077–6085.

[24] S. Zhang, H. Zhao, Y. Wu, Z. Zhang, X. Zhou, X. Zhou, Dual co-matching network for multi-choice reading comprehension, arXiv preprint arXiv:1901.09381 (2019).

[25] Q. Ran, P. Li, W. Hu, J. Zhou, Option comparison network for multiple-choice reading comprehension, arXiv preprint arXiv:1903.03033 (2019).

[26] Y. Yang, W.-t. Yih, C. Meek, Wikiqa: A challenge dataset for open-domain question answering, in: Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 2013–2018.

[27] I. V. Serban, A. García-Durán, C. Gulcehre, S. Ahn, S. Chandar, A. Courville, Y. Bengio, Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus, arXiv preprint arXiv:1603.06807 (2016).

[28] N. Duan, D. Tang, P. Chen, M. Zhou, Question generation for question answering, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 866–874.

[29] 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.

[30] M. Heilman, N. A. Smith, Good question! statistical ranking for question generation, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 609–617.

[31] X. Du, J. Shao, C. Cardie, Learning to ask: Neural question generation for reading comprehension, arXiv preprint arXiv:1705.00106 (2017).

[32] S. Rao, H. Daumé III, Answer-based adversarial training for generating clarification questions, arXiv preprint arXiv:1904.02281 (2019).

[33] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, IEEE Transactions on Knowledge and Data Engineering 34 (1) (2022) 50–70.

[34] J. Wang, W. Xu, X. Fu, G. Xu, Y. Wu, Astral: adversarial trained lstm-cnn for named entity recognition, Knowledge-Based Systems 197 (2020) 105842.

[35] F. Li, Z. Lin, M. Zhang, D. Ji, A span-based model for joint overlapped and discontinuous named entity recognition, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 4814–4828.

[36] Y. Wang, H. Shindo, Y. Matsumoto, T. Watanabe, Nested named entity recognition via explicitly excluding the influence of the best path, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 3547–3557.

[37] L. Liu, B. Ding, L. Bing, S. Joty, L. Si, C. Miao, Mulda: A multilingual data augmentation framework for low-resource cross-lingual ner, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 5834–5846.

[38] B. Ji, Y. Xie, J. Yu, S. Li, J. Ma, Y. Ji, H. Liu, A novel bundling learning paradigm for named entity recognition, Knowledge-Based Systems 248 (2022) 108825.

[39] Z. Liu, G. I. Winata, Z. Lin, P. Xu, P. Fung, Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 8433–8440.

[40] Y. Wang, B. Yu, H. Zhu, T. Liu, N. Yu, L. Sun, Discontinuous named entity recognition as maximal clique discovery, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 764–774.

[41] Y. Shen, X. Ma, Z. Tan, S. Zhang, W. Wang, W. Lu, Locate and label: A two-stage identifier for nested named entity recognition, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 2782–2794.

[42] U. Krcadinac, J. Jovanovic, V. Devedzic, P. Pasquier, Textual affect communication and evocation using abstract generative visuals, IEEE Transactions on Human-Machine Systems 46 (3) (2015) 370–379.

[43] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE transactions on knowledge and data engineering 26 (8) (2013) 1819–1837.

[44] O. Wu, T. Yang, M. Li, M. Li, Two-level lstm for sentiment analysis with lexicon embedding and polar flipping, IEEE Transactions on Cybernetics (2020).

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

[46] M. Kumar, B. Packer, D. Koller, Self-paced learning for latent variable models, Advances in neural information processing systems 23 (2010).

[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[48] Y. Wang, M. Huang, X. Zhu, 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.

[49] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[50] W. He, K. Liu, J. Liu, Y. Lyu, S. Zhao, X. Xiao, Y. Liu, Y. Wang, H. Wu, Q. She, et al., Dureader: a chinese machine reading comprehension dataset from real-world applications, in: Proceedings of the Workshop on Machine Reading for Question Answering, 2018, pp. 37–46.

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