Length-adaptive Neural Network for Answer Selection

Taihua Shao
Science and Technology on Information Systems
Engineering Laboratory
National University of Defense Technology
Changsha, China
shaotaihua13@nudt.edu.cn

Honghui Chen
Science and Technology on Information Systems
Engineering Laboratory
National University of Defense Technology
Changsha, China
chhh0808@gmail.com

Fei Cai
Science and Technology on Information Systems
Engineering Laboratory
National University of Defense Technology
Changsha, China
caifei@nudt.edu.cn

Maarten de Rijke
Informatics Institute
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT
Answer selection focuses on selecting the correct answer for a question. Most previous work on answer selection achieves good performance by employing an RNN, which processes all question and answer sentences with the same feature extractor regardless of the sentence length. These methods often encounter the problem of long-term dependencies. To address this issue, we propose a Length-adaptive Neural Network (LaNN) for answer selection that can auto-select a neural feature extractor according to the length of the input sentence. In particular, we propose a flexible neural structure that applies a BiLSTM-based feature extractor for short sentences and a Transformer-based feature extractor for long sentences. To the best of our knowledge, LaNN is the first neural network structure that can auto-select the feature extraction mechanism based on the input. We quantify the improvements of LaNN against several competitive baselines on the public WikiQA dataset, showing significant improvements over the state-of-the-art.

CCS CONCEPTS
• Information systems → Question answering.

KEYWORDS
Answer selection, Neural network, Question answering.

ACM Reference Format:

1 INTRODUCTION
Answer selection plays a crucial role in a Question Answering (QA) system, which selects the most appropriate answer for a question from a set of candidate answers. Deep learning-based approaches have been well studied for this task [4]. These approaches mainly attempt to generate high-quality sentence embeddings of question and answer, which are then utilized to measure the relevance of a candidate answer to a question.

Previous research has achieved impressive performance on this task. For instance, Tan et al. [6] employ an attentive BiLSTM to measure the relevance of segments in candidate answers for a particular question. Wang and Nyberg [8] apply a stacked BiLSTM-based feature extractor to learn a joint feature vector for question-answer pairs. However, these RNN-based models do not consider an important aspect that has been shown to affect sentence embedding-based tasks, namely the length of a sentence. As the sentence length grows, models will suffer from a long-term dependency problem due to the sequential nature of RNNs. It becomes difficult to learn dependencies between words in distant positions. This, in turn, impacts the quality of embeddings for long sentences, as long-distance interactions between words of may be not be captured [3].

To address this issue, we propose a Length-adaptive Neural Network (LaNN) for the task of answer selection. Specifically, we deploy a hierarchical length-adaptive neural structure to generate sentence embeddings for question and answer sentences, which aims at extracting high-quality sentence features by employing a different neural feature extractor depending on the length of the input sentences. We first generate a word representation for each word in the input sentence by concatenating a frozen word embedding and a fine-tuning word embedding. Then, we propose a flexible neural structure that applies a BiLSTM-based feature extractor for short sentences and a Transformer-based feature extractor for long sentences, respectively. Finally, an attentive pooling layer that takes the interaction between question and answer sentences into consideration is employed to generate the sentence embeddings that are used to measure the relevance of a question and an answer.

We evaluate the performance of the proposed LaNN model against several competitive baselines on a popular QA dataset,
2 APPROACH

Fig. 1 presents an overview of the proposed length-adaptive neural network for answer selection.

2.1 Word representation

Let $s$ be an input sentence (question or answer) with length $L$. To keep as many word features as possible, we produce the word representation corresponding to each word $w_t$ in $s$ not only from frozen pre-trained word embeddings but fine-tuning word embeddings.

Then, we concatenate two word embeddings to form a combined word vector. Finally, we deploy a hidden layer to select useful features from the concatenated word vector. The final representation $r_{w_t}$ (of dimension $D$) of the word $w_t$ is calculated according to:

$$r_{w_t} = \tanh(W_h \cdot (r^f_{w_t} \oplus r^t_{w_t}) + b_h),$$  \hspace{1cm} (1)

where $r^f_{w_t}$ and $r^t_{w_t}$ are corresponding word vectors from the frozen word embeddings and the fine-tuning word embeddings; $W_h \in \mathbb{R}^{D \times D}$ and $b_h \in \mathbb{R}^{D \times 1}$ are network parameters of the hidden layer.

The word representations of $s$ form the word representation matrix:

$$R_W = \begin{bmatrix} r_{w_1} & r_{w_2} & \cdots & r_{w_L} \end{bmatrix},$$  \hspace{1cm} (2)

2.2 Feature extractor

We design a length-adaptive neural network as the feature extractor, making good use of the context information of words in $s$ to generate a high-quality sentence representation for $s$. Most previous neural networks for answer selection [e.g., 1, 6] do not distinguish between input sentences of different lengths. In contrast, we deploy a BiLSTM-based and a Transformer-based feature extractor to deal with sentences of different lengths for generating sentence embeddings. We employ two flags, i.e., $\text{flag}^{\text{tsfm}}$ and $\text{flag}^{\text{lstm}}$, for each feature extractor according to the sentence length $L$ as follows:

$$\begin{cases} \text{flag}^{\text{tsfm}} = 1 & (L < L_{\text{threshold}}) \\ \text{flag}^{\text{lstm}} = 0 & (L_{\text{threshold}} \leq L) \end{cases}$$  \hspace{1cm} (3)

where $L_{\text{threshold}}$ is a preset threshold to judge whether the input sentence $s$ is long or not. The flags will be employed to activate the corresponding feature extractor by multiplying the input word representation matrix $R_W$ with the value of flags as follows:

$$R_W^{\text{tsfm}} = R_W \cdot \text{flag}^{\text{tsfm}}$$  \hspace{1cm} (4)

$$R_W^{\text{lstm}} = R_W \cdot \text{flag}^{\text{lstm}}.$$  \hspace{1cm} (5)

where $R_W^{\text{tsfm}}$ and $R_W^{\text{lstm}}$ are updated input representation matrices for the BiLSTM-based and Transformer-based feature extractors.

For a short sentence, we activate the BiLSTM-based feature extractor by setting $\text{flag}^{\text{tsfm}}$ to 1 and $\text{flag}^{\text{lstm}}$ to 0, which leads to a null representation in the Transformer-based feature extractor. Operations on the $t$-th word representation $r_{w_t}^{\text{lstm}}$ in $R_W^{\text{lstm}}$ are:

$$\begin{align*} f_t &= \sigma(W_r \cdot r_{w_t}^{\text{lstm}} + U_r \cdot h_{t-1} + b_f) \\ i_t &= \sigma(W_i \cdot r_{w_t}^{\text{lstm}} + U_i \cdot h_{t-1} + b_i) \\ C_t &= \tanh(W_c \cdot r_{w_t}^{\text{lstm}} + U_c \cdot h_{t-1} + b_c) \\ C_t^{\ominus} &= f_t \cdot C_{t-1}^{\ominus} + i_t \cdot C_t \\ o_t &= \sigma(W_o \cdot r_{w_t}^{\text{lstm}} + U_o \cdot h_{t-1} + b_o) \\ h_t &= o_t \cdot \tanh(C_t), \end{align*}$$  \hspace{1cm} (6)

where $i$, $f$ and $o$ represent the input gate, the forget gate and the output gate, respectively; $h$ represents the memorized word representation (of dimension $H$); $C$ and $C^\ominus$ are the overall and the present memory, $\sigma$ is the activation function sigmoid; $W \in \mathbb{R}^{H \times D}$, $U \in \mathbb{R}^{H \times H}$ and $b \in \mathbb{R}^{H \times 1}$ are the network parameters, determining the input information, output information and bias, respectively. After deploying an LSTM in two directions, we obtain a BiLSTM-based sentence representation matrix $R_s^{\text{LS}}$ as follows:

$$R_s^{\text{LS}} = h_T^{\ominus},$$  \hspace{1cm} (7)

where $||$ indicates concatenation of two vectors.

For a long sentence, we activate the Transformer-based feature extractor by setting $\text{flag}^{\text{tsfm}}$ to 0 and $\text{flag}^{\text{lstm}}$ to 1. Following [7], we employ a positional encoding to inject sequential information and generate an updated $R_W^{\text{tsfm}}$. After that, a scaled perception function is applied to calculate self-attentive similarity in $R_W^{\text{tsfm}}$:

$$f(R_W^{\text{tsfm}}, R_W^{\text{lstm}}) = O_T \cdot \tanh(W_a \cdot R_W^{\text{tsfm}} + U_a \cdot R_W^{\text{lstm}}),$$  \hspace{1cm} (9)

where $O_T \in \mathbb{R}^{D \times T}$, $W_a \in \mathbb{R}^{D \times D}$ and $U_a \in \mathbb{R}^{D \times D}$ are the attention parameters. Then, the self-attentive sentence representation matrix
\( R^a_d \) is produced by:

\[
R^a_d = R^a_{tsfm} \cdot \text{softmax}\left( f(R^a_{tsfm}, R^a_{tsfm}^T) \right)
\]
\[
= \left( r^a_{w^1}, r^a_{w^2}, \ldots, r^a_{w^n} \right),
\]

where \( d^{tsfm}_W \) is the dimension of the row vector in \( R^a_{tsfm} \); \( d^{tsfm}_W \) aims to scale the softmax function to regulate its value. Finally, \( r^a_{w^i} \) is the \( i \)-th self-attentive word vector in \( R^a_d \).

We then adopt a multi-head mechanism [7] to jointly incorporate information from different representation subspaces. Assuming that there are \( n \) heads in the Transformer-based feature extractor, the \( n \) self-attentive sentence representation matrices will be concatenated to form the output sentence representation matrix \( R^s_{TF} \) as:

\[
R^s_{TF} = R^s_{q} \oplus R^s_{q} \oplus \cdots \oplus R^s_{a^n}
\]
\[
= \left( r^s_{TF_{w^1}}, r^s_{TF_{w^2}}, \ldots, r^s_{TF_{w^j}} \right),
\]

where \( R^s_{q} \) is the \( i \)-th self-attentive sentence representation matrix.

### 2.3 Sentence embedding and answer ranking

After generating the sentence representation matrix in §2.2, we employ an attentive pooling to generate the sentence embeddings \( v_q \) and \( v_a \) for question and answer from their corresponding sentence representation matrices \( R_Q \) and \( R_A \) according to:

\[
G = R_Q^T \cdot U \cdot R_A,
\]
\[
v_q = R_Q \cdot \text{softmax}(\text{ColumnMax}(G)),
\]
\[
v_a = R_A \cdot \text{softmax}(\text{RowMax}(G)),
\]

where \( G \) is the attentive similarity between \( R_Q \) and \( R_A \); \( U \in \mathbb{R}^{D \times D} \) is the attention parameter; \( \text{ColumnMax}(\cdot) \) (or \( \text{RowMax}(\cdot) \)) is a function that returns the max value of a column (or row) vector of a matrix. The relevance of an answer to a question is computed using the cosine similarity of the sentence embeddings [5, 6].

In the training phase, each training instance consists of a question \( q \), a positive answer \( a^+ \) (a ground truth) and a negative answer \( a^- \) (an incorrect answer) randomly sampled from all answers in the training set. We train the neural network for the best training epoch by minimizing the following ranking loss of candidate answers:

\[
\text{loss} = \max \{ 0, m - \cos(v_q, v_{a^+}) + \cos(v_q, v_{a^-}) \},
\]

where \( m \) is a preset margin to judge if a training instance will be terminated or not. By doing so, we can rank the candidate answers according to their relevance towards the corresponding question.

### 3 EXPERIMENTS

#### Model summaries.

We examine the effectiveness of the proposed LaNN model by comparing its performance against the following competitive state-of-the-art baselines: (1) QA-CNN [1]: a CNN-based model that employs a CNN-based feature extractor behind a hidden layer to generate sentence embeddings. (2) QA-BiLSTM [6]: a BiLSTM-based model that employs the BiLSTM-based feature extractor to generate sentence embeddings. (3) AB-LSTM/CNN [6]: an attention-based hybrid model that applies a serial structure to combine CNN and BiLSTM to generate sentence embeddings. LaNN is the answer selection model proposed in this paper.

#### Research questions. (RQ1) Does LaNN beat competitive answer selection models? (RQ2) How does LaNN compare to baseline models for question-answer pairs of different lengths, i.e., short questions with long answers (short-long) and long questions with long answers (long-long)? (As all answers in WikiQA are long, short-short and long-short question-answer pairs are absent.)

#### Dataset and parameters.

The dataset we use to evaluate the performance of LaNN is a publicly available open domain dataset, the WikiQA dataset released in 2015 [9]; statistics of the WikiQA dataset are listed in Table 1.

We set the length threshold to 5 in our experiments, which is close to the average length of questions. Following [5], the pretrained word embedding’s dimension and the size of the hidden layer are set to 300. We pad the sentence length for all questions and answers to 40 [5]. For optimizing the loss, Adam [2] is employed. We train our models in mini-batches and employ exponential decay to vary the learning rate in every epoch. L2 regularization and dropout methods are included in our training process to avoid over-fitting. Table 2 details the main parameters of LaNN.

#### Evaluation metrics.

We view the answer selection task as a ranking problem; it is aimed at ranking the candidate answers according to their relevance towards the question. Hence, following prior research on answer selection [8], we adopt Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) as evaluation metrics.

### 4 RESULTS AND DISCUSSION

#### 4.1 Overall performance

To answer RQ1, we present the results of the three baselines and the LaNN model on the test set of WikiQA. In addition, we investigate the model performance on questions of various types. In particular, we categorize the test set into 5 groups according to the question type, i.e., how, what, who, where, and when. The detailed evaluation scores in terms of MAP and MRR are presented in Table 3.

As shown in Table 3, in general, QA-BiLSTM beats QA-CNN in terms of both metrics. AB-LSTM/CNN shows superiority compared with the three baselines.
to the other two baselines without integrating an attention mechanism. Regarding LaNN, compared to the baselines, it achieves the highest performance in terms of MAP and MRR. In particular, LaNN outperforms QA-BiLSTM with up to 2.74% and 3.45% in terms of MAP and MRR, respectively. The Transformer-based feature extractor in LaNN can help deal with the long-term dependencies problem in the long sentences. Furthermore, the overall MAP and MRR scores of LaNN are increased by up to 1.67% and 2.06% against the best baseline AB-LSTM/CNN. This indicates an LaNN model can help improve the performance of answer selection.

As to questions of different types, LaNN beats three baselines for questions of all types except for type how. The baseline model with the BiLSTM structure, QA-BiLSTM, presents the best results; the BiLSTM-based structure is more effective than the CNN structure as well as the Transformer structure in extracting contextual features hidden in sequential data that is prominent in answering the questions with the type how. The LaNN model achieves its highest improvements in the who group, with an increase of 5.87% and 9.5% against the best baseline AB-LSTM/CNN in terms of MAP and MRR, respectively. The Transformer-based structure employed in the LaNN model, which can deal with long-term dependencies to some extent, is good at extracting features hidden in long-range words for answering the question with the type who.

### 5 CONCLUSIONS AND FUTURE WORK

We propose a length-adaptive neural network (LaNN) for answer selection, which employs a BiLSTM-based feature extractor and a Transformer-based feature extractor to capture global interactions of words to obtain improved sentence embeddings. LaNN can auto-select the neural feature extractor according to the length of the input sentence. Experimental results show LaNN can achieve considerable improvements in terms of MAP and MRR over state-of-the-art baselines. Applying different neural feature extractors to short questions and long answers leads to substantial improvements in answer selection performance. As to future work, we would like to examine the scalability of our proposal by evaluating its effectiveness on other datasets. In addition, we have interest in applying the proposed length-adaptive neural network to other tasks, e.g., text summarization and natural language inference.

## ACKNOWLEDGMENTS

This research was supported by the National Natural Science Foundation of China under No. 61702526, the Defense Industrial Technology Development Program under No. JCKY2017204B064, Ahold Delhaize, the Association of Universities in the Netherlands (VSNU), and the Innovation Center for Artificial Intelligence (ICAI).

## REFERENCES


