Question Difficulty Prediction for READING Problems in Standard Tests

Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, Guoping Hu

†School of Computer Science and Technology, University of Science and Technology of China
{huangzby, zhbk}@mail.ustc.edu.cn, {qiliuql, cheneh}@ustc.edu.cn
‡iFLYTEK Research, {mygao2, siwei, gphu}@iflytek.com
§School of Computer Science and Technology, Anhui University, yusu@iflytek.com

Abstract

Standard tests aim to evaluate the performance of examinees using different tests with consistent difficulties. Thus, a critical demand is to predict the difficulty of each test question before the test is conducted. Existing studies are usually based on the judgments of education experts (e.g., teachers), which may be subjective and labor intensive. In this paper, we propose a novel Test-aware Attention-based Convolutional Neural Network (TACNN) framework to automatically solve this Question Difficulty Prediction (QDP) task for READING problems (a typical problem style in English tests) in standard tests. Specifically, given the abundant historical test logs and text materials of questions, we first design a CNN-based architecture to extract sentence representations for the questions. Then, we utilize an attention strategy to qualify the difficulty contribution of each sentence to questions. Considering the incomparability of question difficulties in different tests, we propose a test-dependent pairwise strategy for training TACNN and generating the difficulty prediction value. Extensive experiments on a real-world dataset not only show the effectiveness of TACNN, but also give interpretable insights to track the attention information for questions.

1 Introduction

In the widely used standard test, such as TOEFL or SAT, examinees are often allowed to retake tests and choose higher scores for college admission (Zhang and Yanling 2008). This rule brings an important requirement that we should select test papers with consistent difficulties to guarantee the fairness. Therefore, measurements on tests have attracted much attention (Boopathiraj and Chellamani 2013).

Among the measurements, one of the most crucial demands is predicting the difficulty of each specific test question, i.e., the percentage of examinees who answer the question wrong (Hontangas et al. 2000). Unfortunately, the question difficulty is not directly observable before the test is conducted, and traditional methods often resort to expertise, such as manual labeling or artificial tests organization (Fuchs et al. 1992). Obviously, these human-based solutions are limited in that they are subjective and labor intensive, and the results could also be biased or misleading.

Figure 1: Two questions of READING problem in tests.

We will illustrate this discovery experimentally. Therefore, it is an urgent issue to automatically predict question difficulty without manual intervention. Fortunately, with abundant tests recorded by automatic test paper marking systems, test logs of examinees and text materials of questions, as the auxiliary information, become more and more available, which benefits a data-driven solution to this Question Difficulty Prediction (QDP) task, especially for the typical READING problems. For example, Figure 1(a) shows an example of a READING problem with 2 questions, and each question contains the corresponding materials of document (TD), question (TQ) and options (TO).

Actually, there are some efforts on text understanding for READING problems, e.g., machine comprehension (Yin, Ebert, and Schütze 2016; Sachan et al. 2015). However, these works could not be directly applied to QDP in standard tests due to the unique challenges in this task. First, READING problems contain multiple parts of text materials (i.e., TD, TQ and TO in Figure 1(a)), which requires an unified way to understand and represent them from a semantic perspective. Second, it is necessary to distinguish the importance of text materials to a specific question, because different questions concern different parts of texts. For example, $Q_1$ in Figure 1(a) concentrates more on the highlighted “blue” sentences while $Q_2$ focuses more on the “green” ones. Third, as shown in Figure 1(b), question ($Q_1$, $Q_2$) difficulties are obviously different in different tests ($T_1$ to $T_3$). This evidence indicates that different questions are incomparable in different tests. E.g., we cannot conclude that $Q_2$ with difficulty 0.6 in $T_1$ is more difficult than $Q_1$ with 0.37
in $T_2$, because the examinees are also different. Thus, it is necessary to take these difficulty biases into consideration for QDP.

To solve QDP with addressing the challenges above, we propose a novel Test-aware Attention-based Convolutional Neural Network (TACNN) framework to automatically predict question difficulty for READING problems before the test is conducted. Specifically, given the historical test logs and text materials of questions, we first design an unified CNN-based architecture to exploit the semantic representations for all text materials (i.e., TD, TQ and TO), so that the multiple parts of texts for each question can be modeled in a common comparable space. Then, we qualify the difficulty contribution of each sentence to one question by utilizing an attention strategy. Next, for training TACNN and generating the difficulty prediction value of each question, we propose a test-dependent pairwise strategy to wipe out the difficulty biases in different tests. Finally, extensive experiments on a large-scale real-world dataset validate both effectiveness and explanatory power of our proposed framework. To the best of our knowledge, this is the first comprehensive data-driven solution to QDP task in standard tests.

### 2 Related Work

Generally, the related work can be classified into the following two categories, i.e., question difficulty studies in educational psychology and text understanding in NLP field.

**Question Difficulty in Educational Psychology.** Question difficulty has been studied for a long time in the field of educational psychology. Some prior works focused on evaluating the possible factors contributed to question difficulty. For example, Beck et al. (1997) held that both question attributes and examinees’ abilities affected question difficulties. Kubinger et al. (2007) found that some attributes were relevant to question difficulty, such as question types, question structures and knowledge depth. Another direction made attempts to leverage examinees’ feedbacks from tests for question evaluation and formed some psychological theories, e.g., classic test theory (CTT) (Alagumalai and Curtis 2005) and cognitive diagnosis assessment (CDA) (DiBello, Roussos, and Stout 2006; Wu et al. 2015). CTT evaluated question difficulty from a statistical perspective while CDA considered it as a parameter obtained from examinees’ responses modeled by a logistic-like function. For predicting question difficulty in practice, traditional solutions often resort to expertise, which heavily relies on manual-labeling for test preparations (Fuchs et al. 1992).

The common limitation of these works is the requirement of manual intervention, which takes a lot of human efforts and expertise. Differently, our study is a complete solution from a data-driven modeling perspective.

**Text Understanding in NLP Field.** One of the most crucial steps in our framework is the understanding and representations of all text materials (Hua et al. 2015; Cui et al. 2016), which aims at extracting textual difficulties for questions in READING problems. This is relevant to many researches in nature language process (NLP), such as question selection (Yu et al. 2014), textual entailment (Bowman et al. 2015) and machine comprehension (Yin, Ebert, and Schütze 2016; Sachan et al. 2015). Generally, existing methods could be classified into two categories: language modeling (Smith et al. 2015) and neural network (NN) (Hermann et al. 2015). In language modeling, some representative works put much emphasis on exploiting syntactic and semantic structures of each question including sentence structures (Bilotti et al. 2007) and lexical grammars (Wang, Smith, and Mitamura 2007). In contrast, NN-based models tried to automatically transform questions into semantic representations. For example, Hermann et al. (2015) proposed a two-layer deep LSTM model for learning text contexts of each question as dynamic ones over the documents. Yin et al. (2016) incorporated attention methods into CNN to model questions from words, phrases to sentences views.

However, all these solutions focused on how hard the machines could choose answers rather than predicting difficulties in standard tests. Therefore, existing solutions could hardly be directly applied to QDP task.

### 3 TACNN Framework

In this section, we first formally introduce the QDP task, and then we introduce the technical details of TACNN. At last, we propose the test-dependent pairwise training strategy.

#### Problem and Study Overview

In this paper, we focus on QDP for READING problems in standard tests, while some other types of problems, such as LISTENING, WRITING and SPEAKING, will be discussed and studied in the future.

**Definition 1 (PROBLEM DEFINITION).** Formally, given a set of questions of READING problems with corresponding text materials including document (TD), question (TQ) and options (TO), and each question $Q_i$ has a difficulty attribute $P_i$ (e.g., 0.6) obtained from test logs (see Table 1), our goal is to leverage the combined instances of question $Q_i$ available (see Table 2) to train a prediction model $M$ (i.e., TACNN), which can be used to estimate the difficulties for questions in the newly-conducted tests.

As shown in Figure 2, our solution is a two-stage framework, which contains a training stage and a testing stage: 1) In the training stage, given test logs of examinees as well as text materials of questions (see Table 2), we propose TACNN to understand and represent all text materials of each question $Q_i$ as corresponding predicted textual difficulty $P_i$. Then considering the difficulty biases shown in

<table>
<thead>
<tr>
<th>TestId</th>
<th>ExamineeId</th>
<th>QuestionId</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>U1</td>
<td>Q1</td>
<td>1</td>
</tr>
<tr>
<td>T1</td>
<td>U2</td>
<td>Q2</td>
<td>1</td>
</tr>
<tr>
<td>T1</td>
<td>U2</td>
<td>Q1</td>
<td>0</td>
</tr>
<tr>
<td>T1</td>
<td>U2</td>
<td>Q2</td>
<td>1</td>
</tr>
<tr>
<td>T2</td>
<td>U3</td>
<td>Q3</td>
<td>1</td>
</tr>
<tr>
<td>T2</td>
<td>U5</td>
<td>Q1</td>
<td>1</td>
</tr>
<tr>
<td>T2</td>
<td>U6</td>
<td>Q3</td>
<td>0</td>
</tr>
</tbody>
</table>

... ... ... ...
Table 2: Examples of question instances combined with test logs and question materials.

<table>
<thead>
<tr>
<th>Difficulty (P)</th>
<th>QuestionId (Q)</th>
<th>TestId (T)</th>
<th>Document (TD)</th>
<th>Question (TQ)</th>
<th>Text Materials</th>
<th>Options (TO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4276</td>
<td>Q1</td>
<td>T1</td>
<td>Larry was on...</td>
<td>In what way...</td>
<td>His daughter had...</td>
<td>His daughter...</td>
</tr>
<tr>
<td>0.4827</td>
<td>Q2</td>
<td>T1</td>
<td>Larry was on...</td>
<td>Why did Larry...</td>
<td>He had become...</td>
<td>His father...</td>
</tr>
<tr>
<td>0.594</td>
<td>Q3</td>
<td>T1</td>
<td>Larry was on...</td>
<td>What can be...</td>
<td>Larry had some...</td>
<td>To take photo...</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>T2</td>
<td>Are you...</td>
<td>Why do people...</td>
<td>They eat too...</td>
<td>Ten-year-old...</td>
</tr>
</tbody>
</table>

Figure 2: The flowchart overview of our work.

Figure 1(b), we propose a test-dependent pairwise strategy for training TACNN. 2) In the testing stage, after obtaining the trained TACNN, for each new question without test logs, we could estimate its difficulty with the available text materials.

Components of TACNN

In this subsection, we will introduce the technical details of TACNN, which learns to represent text materials of questions as predicted difficulties. As shown in Figure 3, TACNN mainly consists of four components, i.e., Input Layer, Sentence CNN Layer, Attention Layer and Prediction Layer. Specifically, Sentence CNN Layer and Attention Layer are the most critical techniques, i.e., the former aims at learning all text materials of each question from a sentence semantic perspective, which is further illustrated in Figure 4; while the latter learns attention representations for each question by qualifying the contributions of its text materials.

Input Layer. The input to TACNN is all text materials of a question \( Q \), i.e., document \( (TD_i) \), question \( (TQ_i) \) and options \( (TO_i) \). Intuitively, \( TD_i \) is formalized with a sequence of sentences \( TD_i = \{s_1, s_2, \ldots, s_M\} \) where \( M \) is the sequence length. \( TQ_i \) and each option in \( TO_i \) are all individual sentences. Moreover, each sentence is combined with a sequence of words \( s = \{w_1, w_2, \ldots, w_N\} \) where \( w_i \in \mathbb{R}^{d_0} \) is initialized by \( d_0 \)-dimensional pre-trained word embedding and \( N \) is the length of sentence. As a result, the document is depicted by a tensor \( TD_i \in \mathbb{R}^{M \times N \times d_0} \) and question \( TQ_i \), or each option in \( TO_i \), is a matrix \( s \in \mathbb{R}^{N \times d_0} \).

Sentence CNN Layer. The second layer is Sentence CNN Layer, where we target at learning each sentence representation from word level. Here, we select CNN-based architecture with the following reasons: 1) By leveraging convolution-pooling operations, CNN is more suitable for capturing dominated information of each sentence from local to global views (Yin, Ebert, and Schütze 2016). This is consistent with the common reading habit that examinees usually understand each sentence by some local key words. 2) CNN can exploit the interactions between words at larger scales and learns the deep comparable semantic representations for sentences. 3) Compared with other deep learning structures, e.g., DNN or RNN, CNN leverages shared convolution filters for training, which reduces the model complexity (Ma, Lu, and Li 2015).

As illustrated in Figure 4, Sentence CNN Layer is a variant of the traditional one (Collobert et al. 2011) that alternates several layers of convolution and \( p \)-max pooling, where each sentence is gradually summarized to a fixed length vectorial representation in final. Here, we introduce the first convolution-pooling operation in detail, and the following deeper ones are defined in the similar way.

Concretely, as shown in Figure 4, given the sentence matrix input \( s \in \mathbb{R}^{N \times d_0} \), the wide convolution operates on a sliding window of every \( k \) words with a kernel \( k \times 1 \). Formally, given the input sentence \( s = \{w_1, w_2, \ldots, w_N\} \), the first convolution operation is set to obtain a new hidden sequence, i.e., \( h^c = \{\tilde{h}^c_1, \ldots, \tilde{h}^c_{N+k-1}\} \), where:

\[
\tilde{h}^c_i = \sigma(G \cdot [w_{i-k+1} \oplus \cdots \oplus w_i] + b), \tag{1}
\]

here, \( G \in \mathbb{R}^{d \times k \times d_0} \), \( b \in \mathbb{R}^d \) are the convolution parameters, and \( d \) is the output dimension. \( \sigma(x) \) is a nonlinear activation function \( ReLU(x) = \max(0, x) \). “\( \oplus \)” is the operation that concatenates \( k \) word vectors into a long vector.

With the convolution process, the sequential \( k \) words are composed to a local semantic representation. Then, we exploit \( p \)-max pooling operation to merge the features from convolution sequence \( h^c \) into a new global hidden sequence, i.e., \( h^{cp} = \{\tilde{h}^{cp}_1, \ldots, \tilde{h}^{cp}_{(N+k-1)/p}\} \), where

\[
\tilde{h}^{cp}_i = \max \left[ h^c_{i-p+1, 1}, \ldots, h^c_{i-p+1, d} \right]. \tag{2}
\]

After that, more layers of convolution-pooling processes are set to gradually summarize the global interactions of words in a sentence and finally reach a vectorial representation one \( s \in \mathbb{R}^{d_1} \), where \( d_1 \) is the output dimension of Sentence CNN Layer.

As a result, the document is transformed into a matrix \( TD_i \in \mathbb{R}^{M \times d_1} \) with \( M \) sentence representations, and texts
After obtaining sentence representations from Sentence CNN Layer, Attention Layer aims at discovering important sentences with high scores as dominant information for a specific question, which is helpful for visualizing the model. In the experiments, we will conduct a deep analysis on attention results to a specific question.

**Prediction Layer.** The last layer is Prediction Layer, where we target at predicting difficulty $\hat{P}_1$ of question $Q_1$ leveraged by the document-attention $DA_1$, the option-attention $OA_1$, and the sentence representation $s^{TQ_1}$ itself. Specifically, we first aggregate them by concatenation operation, then utilize a classical full-connected network (Hecht-Nielsen 1989) to learn the overall difficulty representation $o_i$ and finally predict the difficulty $\hat{P}_1$ by logistic function:

$$
\alpha_i = ReLU(\mathbf{W}_1 \cdot [DA_i \oplus OA_i \oplus s^{TQ_1}] + b_1),
\hat{P}_1 = \text{Sigmoid}(\mathbf{W}_2 \cdot o_i + b_2),
$$

where $\mathbf{W}_1, b_1, \mathbf{W}_2, b_2$ are parameters to tune the network.

**Test-dependent pairwise training strategy.** In this subsection, we propose a pairwise training strategy for TACNN. As shown in Figure 2, after obtaining the predicted textual difficulty from text materials of each question via TACNN, we need to define a proper loss function to make our learning possible in training. In the following, we first straightforwardly define a test-independent loss function and then introduce the test-dependent loss function.

**Test-independent loss function.** Since the question difficulty is not directly observable, we obtain the real difficulty of each question followed by the definition in (Hontangas et al. 2000) from the test logs. For example, in Table 1, the real difficulty of question $Q_1$ could be $P_1 = (1 + 0)/2 = 0.5$. Therefore, we could formulate the QDP task in a supervised way. Intuitively, if we ignore the test characteristics, given all question instances (as shown in Table 2), we could simply...
formulate the test-independent objective function by minimizing the least square loss with a $l_2$-regularization term:

$$J(\Theta) = \sum_{Q_i} (P_i - \mathcal{M}(Q_i))^2 + \lambda_0 \|\Theta_M\|^2,$$ (5)

where $\mathcal{M}$ represents the TACNN that transforms text materials of question $Q_i$ into predicted difficulty $\hat{P}_i$ (Eq. (4)). $\Theta_M$ denotes all parameters in TACNN and $\lambda_0$ is the regularization hyperparameter.

However, as mentioned in Figure 1(b), these calculated difficulties of questions are test-dependent, which means different questions in different tests are incomparable. For example, in Table 1, the difficulty of $Q_1$ is 0.5 and the difficulty of $Q_3$ is 0.33, we cannot get the conclusion that $Q_1$ is more difficult than $Q_3$ because they are in different tests (different TestId) with different examinees. Therefore, if we directly adopt the test-dependent objective function (Eq. (5)), it may introduce some biases into the optimization.

Fortunately, we realize that difficulties of questions in same tests are comparable, e.g., $Q_1$ is more difficult than $Q_2$ in Table 1 because they are both in test $T_1$. Motivating by this, we can model and optimize the difficulty comparison for a pair of questions in same tests by a pairwise strategy.

**Test-dependent pairwise loss function.** Formally, we first construct our test-dependent training triples $\{(T_t, Q_i, Q_j)\}$, as shown in Figure 2, which denotes two different questions $Q_i$ and $Q_j$ in the same test $T_t$. Then the objective function turns to the test-dependent one as:

$$J(\Theta) = \sum_{(T_t, Q_i, Q_j)} ((P^i_t - P^j_t) - (\mathcal{M}(Q_i) - \mathcal{M}(Q_j)))^2 + \lambda_0 \|\Theta_M\|^2,$$ (6)

where $P^i_t$ and $P^j_t$ denote the real difficulties of question $Q_i$ and $Q_j$ in test $T_t$, respectively. In this way, we can learn the model, i.e., TACNN, by directly minimizing the function $J_0$ using AdaDelta (Zeiler 2012).

Then, given $\mathcal{M}$, we could estimate question difficulties of new READING problems only based on the given text materials. Please note that, though we design a test-dependent pairwise strategy for model training, TACNN can be directly adopted for estimating the “absolute difficulty values” (e.g., 0.6) of each new question, since the difficulties of questions are now reflected from the text perspective, such as the words used in the texts. After estimating the difficulties of all the questions in a new test paper, we can decide whether to choose this test paper into the standard test or not.

### 4 Experiments

In this section, we first compare the performance of TACNN against the baseline approaches on QDP task. Then, we make experts comparisons to validate the practical significance of TACNN. At last, we conduct a case study to visualize the explanatory power of TACNN.

#### Dataset Description

The experimental dataset supplied by IFLYTEK is collected from real-world standard tests for READING problems, which contains nearly 3 million test logs of thousands of Chinese senior high schools from the year 2014 to 2016. For preprocessing, we filter the questions without any test log because we cannot obtain their difficulties, and Table 3 shows the basic statistics of the dataset after pruning.

#### Experimental Setup

**Word Embedding.** The word embeddings in Input Layer are trained on a large-scale gigaword corpus using public word2vec tool (Mikolov and Dean 2013) with the dimension 200. Words from READING problems which are not presented in the pre-trained words are initialized randomly.

**TACNN Setting.** In TACNN, we set the maximum length $M$ ($N$) of sentences (words) in documents (sentences) as 25 (40) (zero padded when necessary) according to our observation in Figure 5, i.e., 95% documents (sentences) contains less than 25 (40) sentences (words). Four layers of convolution (three wide convolutions, one narrow convolution) and max-pooling are employed for the Sentence CNN Layer to accommodate the sentence length $N$, where the numbers of the feature maps for four convolutions are (200, 400, 600, 600) respectively. Also, we set the kernel size $k$ as 3 for all convolution layers and the pooling window $p$ as (3, 3, 2, 1) for each max pooling, respectively.

**Training Setting.** We follow (Orr and Müller 2003) and randomly initialize all matrix and vector parameters in TACNN with uniform distribution in the range between $-\sqrt{6/(nin + nout)}$ and $\sqrt{6/(nin + nout)}$, where $nin$ and $nout$ are the numbers of input and output feature sizes of the corresponding matrices, respectively. During the training process, all parameters in TACNN are tuned. Moreover, we set mini batches as 32 for training and we also use dropout (with probability 0.2) in order to prevent overfitting.
Since there have been few prior methods to directly solve QDP task in standard tests, we first introduce some variants of TACNN to highlight the effectiveness of each component of our framework. The details of variants are as follows:

- **CNN**: CNN is a framework with attention-ignored strategy and test-independent loss (Eq. (5)). Here, the attention-ignored strategy means the attention scores \( \alpha \) in Eq. (3) are the same for all sentences in corresponding materials (i.e., documents or options).
- **ACNN**: ACNN is a framework with attention strategy (Eq. (3)) and test-independent loss (Eq. (5)).
- **TCNN**: TCNN is a framework with attention-ignored strategy and test-dependent loss (Eq. (6)).

Besides, we also select HABCNN, whose network architecture is most similar to ours, as another baseline:

- **HABCNN**: A machine comprehension model from (Yin, Ebert, and Schütze 2016) with a kind of CNN and sentence attention. To apply it to QDP task, we adopt its original network architecture and make it a little change by adapting its original softmax based objective to our test-dependent loss (Eq. (6)).

Both TACNN and baselines are all implemented by Theano (Bergstra et al. 2010) and all experiments are run on a Tesla K20m GPU.

### Evaluation Metrics

To measure the performance of TACNN, we first use the widely used Root Mean Squared Error (RMSE) (Salakhutdinov and Mnih 2011) for QDP precision comparison. Besides, we adopt Degree of Agreement (DOA) (Liu et al. 2012) from ranking perspective to measure the percentage of correctly ranked difficulties of question pairs.

We also borrow metrics from educational psychology for evaluation from the test analysis perspective. In educational psychology, for test \( T_i \), the higher positive correlation between real difficulties and predictions of questions, the better performances (Brizuela and Montero-Rojas 2013). Thus, we use the average Pearson Correlation Coefficient (PCC) (Benedy et al. 2009) of all tests to measure the correlation performance. Moreover, we also adopt t-test passing ratio (PR), which is denoted as the percentage of tests which pass t-test at confidence level of 0.05, to evaluate confidence performance.

In summary, the smaller the RMSE is, the better performance the results have. For the other three (DOA, PCC, PR), the larger, the better.

### Experimental Results

#### Overall QDP Results

To observe how the models behave at different data sparsity, we randomly select 60%, 40%, 20%, 10% of standard tests as testing sets, and the rests as training sets, respectively. Note that, to ensure that the questions in testing sets are all new questions and prevent overfitting, we also remove the questions in training sets with same documents which exist in testing sets. Thus, there are no overlaps between the questions in training sets and testing sets.

Figure 6 shows the overall QDP results of all models. We can see that TACNN performs best. Specifically, by optimizing the test-dependent pairwise loss, it beats CNN and ACNN. By qualifying the contributions of texts with the attention strategy, it beats TCNN. Then, HABCNN doesn’t perform as well as TACNN, which indicates that the architecture of HABCNN which aims for the machine comprehension task is unsuitable for QDP task. Last but not least, we can see that models with test-dependent pairwise loss (TACNN, TCNN, HABCNN) perform better than those with test-independent loss (CNN, ACNN). This observation suggests that question difficulties are test-dependent and demonstrates the rationality of pairwise training strategy.

---

**Table 4**: TACNN v.s. Experts on QDP task with PCC metric.

<table>
<thead>
<tr>
<th>Test</th>
<th>TACNN</th>
<th>CNN</th>
<th>HABCNN</th>
<th>TCNN</th>
<th>ACNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0.41</td>
<td>0.21</td>
<td>0.18</td>
<td>0.13</td>
<td>0.38</td>
</tr>
<tr>
<td>T2</td>
<td>0.63</td>
<td>0.68</td>
<td>0.45</td>
<td>0.32</td>
<td>0.52</td>
</tr>
<tr>
<td>T3</td>
<td>0.78</td>
<td>0.70</td>
<td>0.52</td>
<td>0.63</td>
<td>0.28</td>
</tr>
<tr>
<td>T4</td>
<td>0.63</td>
<td>0.40</td>
<td>-0.09</td>
<td>0.07</td>
<td>0.31</td>
</tr>
<tr>
<td>T5</td>
<td>0.53</td>
<td>0.56</td>
<td>0.39</td>
<td>0.32</td>
<td>0.29</td>
</tr>
<tr>
<td>T6</td>
<td>0.47</td>
<td>0.22</td>
<td>0.21</td>
<td>0.01</td>
<td>0.27</td>
</tr>
<tr>
<td>T7</td>
<td>0.81</td>
<td>0.73</td>
<td>0.58</td>
<td>0.29</td>
<td>0.72</td>
</tr>
<tr>
<td>T8</td>
<td>0.77</td>
<td>0.45</td>
<td>0.35</td>
<td>0.45</td>
<td>0.24</td>
</tr>
<tr>
<td>T9</td>
<td>0.81</td>
<td>0.55</td>
<td>0.25</td>
<td>0.54</td>
<td>0.35</td>
</tr>
<tr>
<td>T10</td>
<td>0.76</td>
<td>0.57</td>
<td>0.40</td>
<td>-0.13</td>
<td>0.72</td>
</tr>
<tr>
<td>T11</td>
<td>0.90</td>
<td>0.77</td>
<td>0.44</td>
<td>0.57</td>
<td>0.59</td>
</tr>
<tr>
<td>T12</td>
<td>0.60</td>
<td>0.62</td>
<td>0.59</td>
<td>0.73</td>
<td>0.60</td>
</tr>
<tr>
<td>Avg</td>
<td>0.68</td>
<td>0.54</td>
<td>0.36</td>
<td>0.33</td>
<td>0.44</td>
</tr>
<tr>
<td>Std</td>
<td>0.14</td>
<td>0.18</td>
<td>0.19</td>
<td>0.26</td>
<td>0.17</td>
</tr>
</tbody>
</table>

---

**Figure 6**: Overall performance on the task of QDP.
Figure 7: Attention visualization of the document material for question $Q_2$ in Figure 1(a), where too long sentences are truncated with ‘...’. The left bar charts denote the distribution of attention scores over sentences in the document.

Experts Comparison. To demonstrate the practical significance of TACNN, we select 12 standard tests and invite 7 experts (high school teachers) who are familiar with READING problems to do QDP task manually. In detail, each selected test contains 4 READING problems and 16 questions. Experts (denoted as Ep1 to Ep7) are asked to answer the questions and then value the difficulties individually. Furthermore, we average their predictions which is denoted as EpAvg. Thus we totally obtain 8 experts’ predictions. Following educational psychology, we use PCC to assess the correlations between all predictions and real difficulties in tests. All the results are shown in Table 4.

As we can see, TACNN outperforms all experts in most cases, which means predictions from TACNN are the most correlated to the practices. Besides, we also observe that predictions from experts are not always consistent. Specifically, for each test, there are some experts doing the QDP task well (e.g., Ep2 in T3) but others may fail (e.g., Ep5 in T3), because they all make the predictions by subjective judgments, which are hardly of the same minds. Thus, experts’ predictions may be misleading sometimes.

Case Study. One important characteristic of TACNN is its explanatory power to distinguish the difficulty contributions of text materials to a specific question, i.e., the attention scores $\alpha$ in Eq. (3). Figure 7 shows the attention scores of each sentence in the document for question $Q_2$ (“Why did Larry have to stay in a cage underwater sometimes?”) in Figure 1(a). We can see that four highlighted ‘red’ sentences in the document have the highest attention scores, indicating they contribute the most difficulty to $Q_2$. This visualization hints that TACNN provides a good way for a question to capture key information for model explanations.

Discussion. From the experimental results, we can observe that TACNN works well for QDP task in standard tests. Furthermore, the case study shows that our framework could give interpretive results. In the future, there are still some directions for further studies. First, we will make our efforts to design a more efficient learning algorithm for TACNN. Second, we are also willing to extend TACNN to solve QDP task in other types of problems in English tests, such as LISTENING, WRITING (Leki, Cumming, and Silva 2010) and SPEAKING, and also in other subjects, e.g., MATH.

5 Conclusions

In this paper, we proposed a novel Test-aware Attention-based Convolutional Neural Network (TACNN) framework to solve QDP task for READING problems in standard tests. Specifically, we first designed a CNN-based architecture for exploiting sentence representations for the text materials of questions. Then, we qualified the contributions of sentences to question difficulties by an attention strategy. Finally, we proposed a test-dependent pairwise strategy for training TACNN and generating the difficulty prediction values. The experimental results on a real-world dataset clearly demonstrated both the effectiveness and explanatory power of our proposed framework. We hope this work could lead to more studies in the future.

6 Acknowledgements

The authors thank National Education Examinations Authority for providing valuable evaluations for our experiments. This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the National Natural Science Foundation of China (Grants No. 61403358 and 61672483), and the National High Technology Research and Development Program of China (Grant No. 2015AA015409). Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association of CAS (No. 2014299).

References


Boopathiraj, C., and Chellamani, K. 2013. Analysis of test items on difficulty level and discrimination index in the test


