Automatic Scoring Method for Descriptive Test Using Recurrent Neural Network

Keiji Yasuda
KDDI Research
Garden Air Tower, 3-10-10,
Iidabashi, Chiyoda-ku, Tokyo
102-8460 Japan
ke-yasuda@kddi-research.jp

Izuru Nogaito
KDDI Research
2-1-15, Ohara, Fujimino city,
Saitama, 356-8502 Japan
iz-nogaito@kddi-research.jp

Hiroyuki Kawashima
KDDI Research
Garden Air Tower, 3-10-10,
Iidabashi, Chiyoda-ku, Tokyo
102-8460 Japan
hi-kawashima@kddi-research.jp

Hiroaki Kimura
KDDI Research
Garden Air Tower, 3-10-10,
Iidabashi, Chiyoda-ku, Tokyo
102-8460 Japan
ha-kimura@kddi-research.jp

Masayuki Hashimoto
KDDI CORPORATION
Garden Air Tower, 3-10-10,
Iidabashi, Chiyoda-ku, Tokyo
102-8460 Japan
mu-hashimoto@kddi.com

ABSTRACT

In this paper, we propose an automatic evaluation method for the descriptive type test. The method is based on Recurrent Neural Networks trained on a non-labeled language corpus and manually graded students' answers. The experimental results show that the proposed method is the second best result among five conventional methods, including BLEU, RIBES, and several sentence-embedding methods. And, the proposed method gives the best performance among several sentence embedding methods.

Keywords

RNN, LSTM, Language Model, Essay Scoring

1. INTRODUCTION

Twenty-first-century skills are advocated in the educational field. Compared to traditional knowledge-based education evaluated by multiple-choice tests, the evaluation of twenty-first-century skills is very difficult. A descriptive test is one solution to the problem, although the cost of scoring is prohibitive. In this paper, we propose a method to automatically score descriptive type tests to solve the problem stated above. The method uses long short-term memory (LSTM) recurrent neural networks (RNN) to score the answers written in natural language. The method requires two kinds of data sets.

One is a large language corpus used for pre-training of RNN. As pre-training, the RNN-based language model is trained using the corpus. A vector given by a hidden layer in the networks is thought to embed the meaning of processed sentences. Thus, the proposed method calculates the similarity between two vectors given by processing model answers and student answers on RNN. The other data set is a small labeled corpus that consists of model answers, student answers, and manually annotated scores of student answers. The labeled corpus is used for training of the RNN.

2. PROPOSED METHOD

The RNN framework used in the paper is shown in Fig. 1. As shown in the figure, the proposed method uses two kinds of corpora and two kinds of training parts. They are the pre-training of word embedding and the main training of the LSTM-type RNN [3].

Here, we express the sentence ($\mathbf{s}$) as the sequence of words $\mathbf{s} = w_1, \cdots, w_t, \cdots, w_T$. The word-embedding part projects the input word of time t (w_t) to high-dimension vector $x_{w_t} \in \mathbb{R}^{d_w}$ as follows:

$$\mathbf{x}_{w_t} = \mathbf{E}^{\mathrm{T}} \mathbf{w}_{w_t} \tag{1}$$

where $w_{w_t} \in \mathbb{R}^{|V|}$ is the one-hot vector of w_t and $\mathbf{E} \in \mathbb{R}^{|V| \times d_w}$ is the lookup table. $x_{w_t}$is used as the input for the LSMT part. The LSTM consists of four components: the forget gate ($\mathbf{f}_t$), input gate ($\mathbf{i}_t$) and output gate ($\mathbf{o}_t$), and the memory state ($\mathbf{c}_t$). These real-valued vectors are calculated by the following formulas:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_{w_t} + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_{w_t} + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \\
\mathbf{o}_t &= \sigma(\mathbf{W}_i \mathbf{x}_{w_t} + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_{\tilde{c}} \mathbf{x}_{w_t} + \mathbf{U}_{\tilde{c}} \mathbf{h}_{t-1} + \mathbf{b}_{\tilde{c}}), \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t
\end{aligned} \tag{2}$$

where $\mathbf{W}$ and $\mathbf{U}$ are weight matrices, and $\mathbf{b}$ is the bias vector. $\sigma(\cdot)$ and $\tanh(\cdot)$ are an element-wise sigmoid function

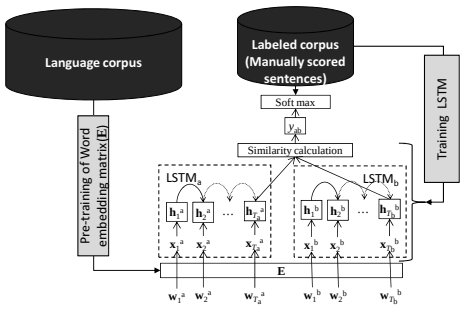

Figure 1: Framework of the proposed method.

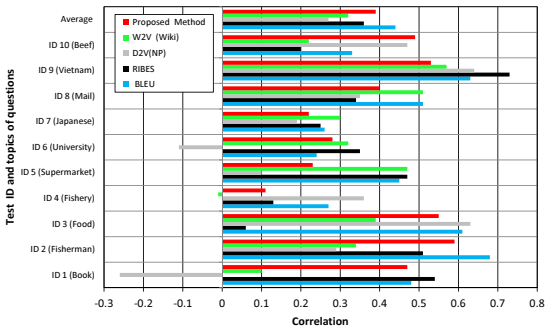

Figure 2: Experimental results.

and a hyperbolic tangent function, respectively. Using these vectors, hidden-layer vector ($\mathbf{h}_t \in \mathbb{R}^{d_s}$) is calculated as follows:

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{3}$$

where $\odot$ is element-wise multiplication. The main training part requires a labeled corpus that consists of model answers, the students' answers, and manually scored results of the students' answers. By using the labeled corpus, the second training part tunes the LSTM whose network configuration was proposed by Mueller et al. [1]. Using pre-trained word-embedding matrix $\mathbf{E}$ from the first training part, LSTM parameters are trained as follows.

First, randomly initialize LSTM parameters in Eq. 2. Then, duplicate the initialized LSTM ($\mathrm{LSTM_a}$ and $\mathrm{LSTM_b}$ in Fig. 1). One of them is used to process the student's answer and the other is used to process the model answer. We regard the hidden-layer vector of the sentence end as sentence embedding. To calculate the sentence similarity between the student's answer and the model answer, we add a new unit between the hidden layers. The unit calculates the L1 norm based on the similarity between the two sentence embeddings ($\mathbf{h}^{\mathrm{a}}_{T_{\mathrm{a}}}$ and $\mathbf{h}^{\mathrm{b}}_{T_{\mathrm{b}}}$ in Fig. 1) by using the following formula [1]:

$$\begin{aligned} g(\mathbf{h}^{\mathrm{a}}_{T_{\mathrm{a}}}, \mathbf{h}^{\mathrm{b}}_{T_{\mathrm{b}}}) &= \exp(-\|\mathbf{h}^{\mathrm{a}}_{T_{\mathrm{a}}} - \mathbf{h}^{\mathrm{b}}_{T_{\mathrm{b}}}\|_1) \\ &= \exp\left(-\sum_{i=1}^{d_s} \left|h^{\mathrm{a}}_{T_{\mathrm{a}}} - h^{\mathrm{b}}_{T_{\mathrm{b}}}\right|\right) \end{aligned} \tag{4}$$

The similarity calculation is performed only when both sentence pairs have been processed by the LSTM. Using the similarity calculated by Eq. 4 and the manually evaluated score, the deviation is back propagated to tune the LSTM weights. Here, we restrict the parameters of $\mathrm{LSTM_a}$ and $\mathrm{LSTM_b}$ to the same values.

3. EXPERIMENTS

The labeled corpus consists of 10 descriptive type questions and their answers. For each question, around 20 answers are manually scored. Additionally, there are also four model answers for each question. For the pre-training of the word-embedding matrix, we use a Mainichi newspaper corpus.

Since the size of the labeled corpus is very small, we carry out a leave-one-out cross-validation test for each question. The cross-validation is carried out only for student answers. The same model answers are used for training and evaluation. The LSTM in the paper can only process a pair of one student answer and one model answer at the same time. Thus, all combinations of student answers and model answers in the training set are used for training. For the scoring of the test set, we calculate the average score of several model answers. The evaluation measure is the correlation coefficients between the manual and the automatic scoring results.

Fig. 2 shows the experimental results. As baseline results, we show the results of BLEU, RIBES, and the Doc2Vec (D2V) cosine similarity method with the NewsPaper(NP) corpus and Wikipedia(Wiki) corpus by referring to the conventional research[2]. As shown in the figure, the proposed method never gives a negative correlation coefficient. Meanwhile the conventional sentence-embedding-based methods give negative correlation coefficients. Additionally, the proposed method gives the best results on average among sentence-embedding methods, which are two kinds of D2V and the proposed method. Compared to all methods, the proposed method offers the second-best performance.

4. CONCLUSIONS AND FUTURE WORKS

We proposed the LSTM-based automatic scoring method for descriptive tests. We carried out experiments using actual learning logs. According to the experimental results, the proposed method gives the best performance among several sentence-embedding methods, and the second-best results among five methods including BLEU and RIBES.

5. ACKNOWLEDGMENTS

This work used model answers, students' answers, and scoring data forms from the Lojim School. (http://lojim.jp/).

6. REFERENCES

[1] J. Mueller et al. Siamese recurrent architectures for learning sentence similarity. In Proc. of AAAI, pages 2786–2792, 2016.

[2] I. Nogaito et al. Study on automatic scoring of descriptive type tests using text similarity calculations. In Proc. of EDM, pages 616–617, 2016.

[3] M. Sundermeyer et al. LSTM neural networks for language modeling. In Proc. of Interspeech, pages 194–197, 2012.