Assessing Student Contributions in Wiki-based Collaborative Writing System

Tianyu Hu
University of Science and Technology of China
hty98@mail.ustc.edu.cn

Guangzhong Sun
University of Science and Technology of China
gzsun@ustc.edu.cn

Zhongtian Xu
University of Science and Technology of China
xuzt@mail.ustc.edu.cn

ABSTRACT

In recent years, Wiki has been proved effective for collaborative learning in modern education. As a typical collaborative writing system , Wiki empowers students in generating, modifying and structuring their own contents. Some courses may include these collaborative assignments like writing a wiki page as part of assessment. But for teachers, it is difficult to assess the quality of student contributions, because the final result of project is made up of edits from different students. In this paper, we propose a content-based model, OSEAN(Order-Sensitive Edit Assessing Network) to better address this problem. OSEAN can represent and predict students edits' quality by extracting semantic features from edit pairs. Experiment results show that OSEAN has the highest AUPRC on Wikipedia edit quality classification task in all tested methods. Furthermore, OSEAN can handle reversed edit pairs correctly, which often happens when one student undoes previous student's edit.

Keywords

Natural Language Processing,Assessment,Collaborative Learning,Sequence Modeling,Wikipedia,Crowdsourcing

1. INTRODUCTION

In recent years, the use of modern information and communication technologies in education has been widely studied[10]. Thanks to the rapid development of web technology, higher level of collaborative learning becomes easier. Among these web applications, wiki attracted attention for enabling students work together. According to the definition on Wikipedia, wiki is a knowledge base website on which users collaboratively modify and structure content directly from a web browser. These inherent characteristics of wiki technology encourage students collaborate to create their own contents[3].

However, assessing student contributions in a wiki project can be difficult. This is because that students not only add contents to the project, but also revise or delete contents which are added by others. Since reprocessability plays a key role in evaluation of student works[6], we should assess student contributions from the entire process of wiki project. If teachers only evaluate everyone's contribution from the final state of the project, then some important behavior information can be lost. Figure 1 gives a example of page revision history. In this work, we care about the quality of students contribution in the project, so we need to evaluate the quality of each edit. A wiki project usually consists of many edits, which brings a lot of works to teachers. Therefore, we want to evaluate the quality of edits in an automated way. To predict new edits' quality, two types of methods are proposed. Content-based methods extract features from the content of edits. ORES[5] and Stiki[11], web services provided by Wikimedia team, use linguistic features to compute the probability that a specific edit is damaging. StRE[8] utilizes deep neural network and achieves a high accuracy. On the other hand, content-independent methods, e.g. Interank[12], treat edit as the interaction between user and project(page).

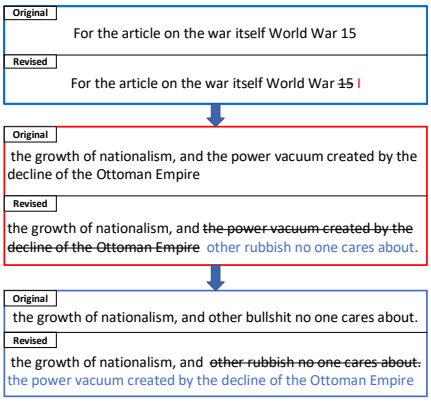

Figure 1: Example of revision history from Wiki page: Causes of World War I. We select 3 continuous versions and compare the differences. Edit 1 fixed an error in the page. Edit 2 deleted some words and added some offensive words. Edit 3 did a revert operation to eliminate vandalism information introduced by revision 2.

While each edit is a pair of sequences before and after an edit, a new question arises: Does the order of pair matter? The order of edit pairs represents the direction of contents

Tianyu Hu, Guangzhong Sun and Zhongtian Xu "Assessing Student Contributions in Wiki-based Collaborative Writing System" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 615 - 619

evolution. If a model can truly predict the quality of an edit, it should generate a opposite label if we reverse the edit pair.

To better predict new edits' quality and handle the order of edit pair. In this work, we propose OSEAN(Order-Sensitive Edit Assessing Network), a content-based edit quality prediction model. OSEAN extracts each dissimilar part of two sentences and learn the vector representations for two parts. To handle the order of edit pair, we utilize the subtract result between two parts as the final representation of the entire edit.

2. METHODOLOGY

2.1 Problem Formulation

An edit $\mathrm{P} = \{\mathrm{S}, \mathrm{T}\}$ on a particular page is a pair of original sentence S and revised sentence T. Each sequence is represented as a fixed length character sequence.

$$\mathrm{S} = \{\mathrm{S}_1, \mathrm{S}_2, ..., \mathrm{S}_\mathrm{M}\}$$
$$\mathrm{T} = \{\mathrm{T}_1, \mathrm{T}_2, ..., \mathrm{T}_\mathrm{M}\}$$

where M is the length that can be manually set. Our task is to find a page-specified scoring function that maps each edit to a binary label:

$$f_{page} : \mathrm{P} \rightarrow \mathrm{L}, \mathrm{L} \in \{0, 1\} \tag{1}$$

2.2 Model Architecture

Figure 2 gives an overview of OSEAN. We will introduce each steps in the model below.

Character Embedding. The first layer performs a character-level look-up where each character is represented as a d-dimension vector. The edit pair is converted to two matrices of dimension $m \times d$.

Convolution Step. After the character-level embedding, the sequences of embedded characters is provided as inputs of convolution layer, which computes an 1-D convolution over the embedded sequences. A convolution operation involves a filter with size h :

$$c_i = \tanh(w_c \cdot x_{i:i+h-1} + b_c) \tag{2}$$

As a result, each sentence is represented as a feature map of dimension $l \times d$, where $l = m - h + 1$.

Dissimilar Part Extraction. Since an edit is changes of page contents, the dissimilar part of two sequences should have higher weights on qualities. We utilize the method from [9]. In our model, the semantic unit of the sequence is the combinations of characters after the convolution operation, and we only care about the dissimilar part. To determine which part is *dissimilar*, we need to check whether a unit is semantically covered by another sequence.

First, we compute the similarity matrix $\mathrm{A}_{L \times L}$ for feature maps $\mathbf{C}_\mathrm{S}$ and $\mathbf{C}_\mathrm{T}$ after the convolution step, each element $a_{i,j} \in \mathrm{A}$ is the cosine similarity between unit $\mathbf{C}_{\mathrm{S},i}$ and $\mathbf{C}_{\mathrm{T},j}$.

$$a_{i,j} = \frac{\mathbf{C}_{\mathrm{S},i}^\top \mathbf{C}_{\mathrm{T},j}}{\|\mathbf{C}_{\mathrm{S},i}\| \|\mathbf{C}_{\mathrm{T},j}\|} \tag{3}$$

Then we use the similarity matrix to calculate the semantic cover of $\mathbf{C}_{\mathrm{S,i}}$ by combining all units in the other sequence $\mathbf{C}_\mathrm{T}$.

$$\mathrm{cover}(\mathbf{C}_{\mathrm{S},i}, \mathbf{C}_\mathrm{T}) = \frac{\sum_{j=0}^{L} a_{i,j} \mathbf{C}_{\mathrm{T},j}}{\sum_{j=0}^{L} a_{i,j}} \tag{4}$$

The result $\hat{\mathbf{C}}_{\mathrm{S},i} = \mathrm{cover}(\mathbf{C}_{\mathrm{S},i}, \mathbf{C}_\mathrm{T})$ can be used to calculate the proportion α of unit $\mathbf{C}_{\mathrm{S},i}$ that is present in the other sequence. The value of α is the cosine similarity α of $\mathbf{C}_{\mathrm{S},i}$ and $\hat{\mathbf{C}}_{\mathrm{S},i}$. So the dissimilar part's can be defined as $1 - \alpha$. The dissimilar part $\mathbf{D}_{\mathrm{S},i}$ for feature map unit $\mathbf{C}_{\mathrm{S},i}$ is:

$$\alpha_i = \frac{\mathbf{C}_{\mathrm{S},i}^\top \hat{\mathbf{C}}_{\mathrm{S},i}}{\|\mathbf{C}_{\mathrm{S},i}\| \|\hat{\mathbf{C}}_{\mathrm{S},i}\|} \tag{5}$$

$$\mathbf{D}_{\mathrm{S},i} = (1 - \alpha_i)\mathbf{C}_{\mathrm{S},i} \tag{6}$$

After performing the above calculations for all units in $\mathbf{C}_\mathrm{S}$ and $\mathbf{C}_\mathrm{T}$, we get two dissimilar parts $\mathbf{D}_\mathrm{S}$ and $\mathbf{D}_\mathrm{T}$.

Edit Representation. We use a weight-sharing fully connected layer(FCL) to generate representation vectors $\mathbf{E}_\mathrm{S}, \mathbf{E}_\mathrm{T}$ for each sequence. To obtain the final representation $\mathbf{E}_\mathrm{final}$ for the whole edit, we perform a **subtract** operation on $\mathbf{E}_\mathrm{S}$ and $\mathbf{E}_\mathrm{T}$. The edit vector E_{final} is used for quality classification with a sigmoid activation.

$$\mathbf{E}_\mathrm{S} = \mathbf{W}_0 \mathbf{D}_\mathrm{S} + \mathrm{b}_0, \ \mathbf{E}_\mathrm{T} = \mathbf{W}_0 \mathbf{D}_\mathrm{T} + \mathrm{b}_0 \tag{7}$$

$$\mathbf{E}_\mathrm{final} = \mathbf{E}_\mathrm{S} - \mathbf{E}_\mathrm{T} \tag{8}$$

$$r = \mathrm{sigmoid}(\mathbf{W}_1 \mathbf{E}_\mathrm{final} + \mathrm{b}_1) \tag{9}$$

Here, r is considered to be the possibility that the edit P to be a beneficial edit.

2.3 Order of Edit Pair

Consider an edit $\mathrm{P} = (\mathrm{S}, \mathrm{T})$, we assume P to be a beneficial edit and labeled as **1**. If we reverse the order of the edit pair, the label of the reversed edit $\mathrm{P}' = (\mathrm{T}, \mathrm{S})$ should also be flipped, meaning P' has a label **0**. This is because the reverted operation on a beneficial edit should be considered to be a damaging edit. If the order of the pair can not be handled correctly, the model is very likely to classify two opposite edit P and P' to be the same label.

We give the definition of *order-sensitive* here:

Definition 1 (order-sensitive)**.** *A model is order-sensitive if for most edit pairs, it satisfies: the model gives two opposite labels for edit* P *and its reversed version* P'*.*

Obviously, an ideal edit quality prediction model should be order-sensitive. Our proposed model is *perfectly* order-sensitive under ideal conditions which can be proven mathematically and also performed well in the experiment:

3. EXPERIMENTS

In this section, we conduct experiments to answer following questions:

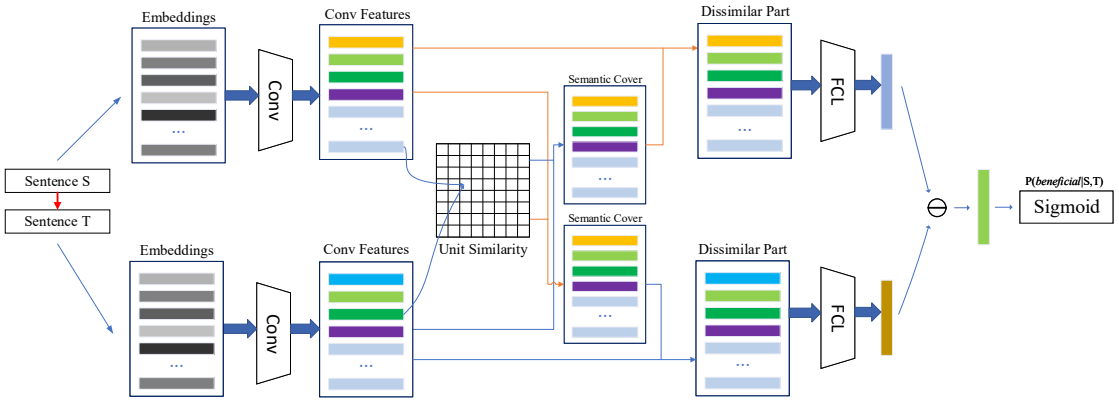

Figure 2: An overall architecture of OSEAN(Order-Sensitive Edit Assessing Network)

RQ1 Can our proposed model outperform state-of-the-art edit quality prediction methods?

RQ2 If the order of edit pairs is reversed, how the performance of all experimented methods will change? Can our model still maintain the best performance?

RQ3 If we add the order information to model training process by performing data augmentation, will model performance get improved?

3.1 Experiments Setup

3.1.1 Dataset

We evaluate model's performance on the data extracted from Wikipedia page revision histories. As Wikipedia is the most widely used collaborative writing system in the world, experiments on this system can verify the effectiveness of our model. Page histories are divided into three categories:

1. CS: Pages containing top 147 pages with the highest number of edits related to computer science in English Wikipedia as of June 2017[8].
2. EN/ZH: Pages containing top 68/55 pages in the whole English/Chinese Wikipedia as of June 2019.

The number of samples in each category is reported in 1.

3.1.2 Computation of Edit Quality and Label

The basic idea is that if changes introduced by an edit is preserved in several subsequent edit, then the edit is considered to be beneficial. Otherwise, if the changes is reverted, then the edit is damaging. [1] and [2] give a formula to compute the proportion of preserved changes. We follow the approach and use a average value to compute the edit quality.

Consider a particular page and denote its k-th revision (i.e., the state of the article after the k-th edit) as v_k. Let $d(u, v)$ be the Levenshtein distance[7] between two sentences. We define the quality of edit k from the perspective of the article's state after $\ell \geq 1$ subsequent edits as:

$$q_{k|\ell} = \frac{d(v_{k-1}, v_{k+l}) - d(v_k, v_{k+l})}{d(v_{k-1}, v_k)} \tag{10}$$

Samples	CS	EN	ZH	Total
# Total	2377732	285365	122748	2785845
# q $\geq$ 0	1402596	190924	88621	1682141
# q $<$ 0	975136	94441	34127	1103704

Table 1: Number of samples for each category in dataset

We compute the average value over several future revisions:

$$q_k = \frac{1}{L} \sum_{\ell=1}^{L} q_{k|\ell} \tag{11}$$

We set $L = 10$ to compute the final edit quality in data preprocessing. Each edit's quality is automatically computed and labeled as *damaging* if the quality score $q < 0$, and labeled as *beneficial* if $q \geq 0$.

3.1.3 Competing Approaches

We compare OSEAN with some existing methods:

Average The average approach always outputs the ratio of good edit on the training set as the predict probability.

ORES The Objective Revision Evaluation Service (ORES)[4, 5] is an open-source classifier system developed by researchers at the Wikimedia Foundation.

Interank Interrank[12] uses matrix factorization method to learn editor's ability and page's difficulty based on the page's edit history.

StRE StRE(Self Attentive Revision Encoder)[8] is a deep learning based method which combines word level signals as well as character level signals.

ABCNN Attention Based Convolutional Neural Network (ABCNN)[13] integrates attention into CNNs for general sentence pair modeling tasks. We use ABCNN-2 for our edit classification task.

3.1.4 Evaluation

To compare the performance of models, we set up a classification task to predict if an edit is beneficial or not. For

Model	CS	EN	ZH	Total
Average	0.733	0.714	0.814	0.745
Interank	0.448	0.427	0.352	0.436
ORES	0.832	0.852	0.838	0.834
StRE	0.898	0.877	0.884	0.890
ABCNN	0.899	0.912	0.938	0.905
OSEAN	**0.946**	**0.945**	**0.952**	**0.947**

Table 2: Results on Wikipedia dataset

Model	On Test			On Train		
	Ori-Test	Rev-Test	Diff	Ori-Train	Rev-Train	Diff
Average	0.745	0.305	-0.440	0.745	0.270	-0.475
ORES	0.835	0.316	-0.519	0.931	0.286	-0.645
StRE	0.890	0.404	-0.486	0.923	0.345	-0.578
ABCNN	0.905	0.497	-0.408	0.924	0.450	-0.474
OSEAN	**0.948**	**0.755**	**-0.193**	**0.993**	**0.957**	**-0.036**

Table 3: Results for reversed pair experiment. Ori-* denotes the original set, Rev-* denotes the reversed set.

each example, we compute the quality score based on the revision history and assign each example a binary label.

For each particular page, we split the edits on the page randomly into train/validation/test set with ratio 80%/10%/10% and train models. Page-specific models are evaluated and we use the average AUPRC in each category as the final metric which is consistent to previous works[12, 8].

3.2 Basic Experiment (RQ1)

We evaluate OSEAN on the original test set to answer the first question. Table 2 presents the average AUPRC value for each category in original test set. OSEAN has the highest AUPRC and is 4.6% higher than the next-best method, proving the effectiveness of our proposed model.

3.3 Reversed Pair Experiment (RQ2)

In this experiment, we use the same train and validation set as before. For test set, we design two settings:

1. On Test : Trained models are evaluated on original and reversed *test* set.

2. On Train: Trained models are evaluated on original and reversed *train* set.

A reversed dataset is generated by reversing every edit pair and flipping the labels in the original set. According to the definition, an order-sensitive model should have similar performance on original and reversed set. Thus, the difference in AUPRC can be used as a criterion to determine whether the model is order-sensitive. We use average AUPRC of all pages as metric.

Results. Experiment results are reported in Table 3. Interank model is not tested because the reversed sample is anonymous which can not be processed by Interank. Performance of all models drops when classifying reversed pairs. OSEAN has the smallest decline which is 52.7% lower on test and 92.4% lower on train than the next-best method. OSEAN has the smallest performance decline and highest AUPRC on reversed set in both settings, proving that our model can handle reversed edit pairs correctly.

3.4 Training with Augmentation (RQ3)

In this experiment, we use training set with data augmentation to train models. For each example $P = (S, T)$ with label ℓ in training set, we add a reversed example $P^{'} = (T, S)$ with the opposite label $\ell^{'}$ to the training set. Models are trained on augmented training set and evaluated on both original and reversed test set. We train models with five different cases (i.e. when 0%/25%/50%75%100% of reversed training pairs are added). We want to know if data augmentation allows models to learn the information of pair order and empowers models to be order-sensitive.

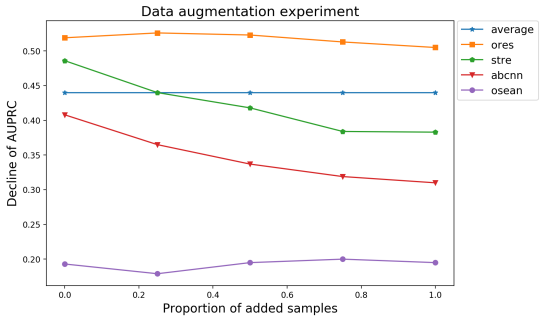

Figure 3: Results on decline of AUPRC with different proportion of data augemented.

Results. Performance decline with different rates of data augmentation is reported in Figure 3. As more data is added, the performance gap between the original and reversed test set is also declined. The narrowing of the gap proves that data augmentation can indeed make models more order-sensitive. However, even with 100% data augmentation, the performance gap for all baseline methods is still large, and gap for OSEAN is 37.4% lower than the next-best method.

4. CONCLUSION

In this paper, we present OSEAN, a content-based model for assessing edit quality in wiki-based writing system. Our method utilizes the convolution network to find semantic differences between previous and revised sentences, which can represent an edit. Experimental results on page revision histories from Wikipedia demonstrate that our model can effectively predict new edits' quality. Therefore, we can more accurately determine the quality of student contributions in the project.

5. ACKNOWLEDGMENT

This work is supported by Youth Innovation Promotion Association of CAS. Guangzhong Sun is the corresponding author of this work.

6. REFERENCES

[1] B. T. Adler and L. De Alfaro. A content-driven reputation system for the wikipedia. In *Proceedings of the 16th international conference on World Wide Web*, pages 261–270. ACM, 2007.

[2] B. T. Adler, L. De Alfaro, I. Pye, and V. Raman. Measuring author contributions to the wikipedia. In *Proceedings of the 4th International Symposium on Wikis*, page 15. ACM, 2008.

[3] M. Cole. Using wiki technology to support student engagement: Lessons from the trenches. *Computers & education*, 52(1):141–146, 2009.

[4] A. Halfaker and R. S. Geiger. Ores: Lowering barriers with participatory machine learning in wikipedia. *arXiv preprint arXiv:1909.05189*, 2019.

[5] A. Halfaker and D. Taraborelli. Artificial intelligence service "ores" gives wikipedians x-ray specs to see through bad edits, 2015.

[6] W. He and L. Yang. Using wikis in team collaboration: A media capability perspective. *Information & Management*, 53(7):846–856, 2016.

[7] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[8] S. Sarkar, B. P. Reddy, S. Sikdar, and A. Mukherjee. Stre: Self attentive edit quality prediction in wikipedia. *arXiv preprint arXiv:1906.04678*, 2019.

[9] Z. Wang, H. Mi, and A. Ittycheriah. Sentence similarity learning by lexical decomposition and composition. *arXiv preprint arXiv:1602.07019*, 2016.

[10] M. Warschauer. Computer-mediated collaborative learning: Theory and practice. *The modern language journal*, 81(4):470–481, 1997.

[11] A. G. West, S. Kannan, and I. Lee. Stiki: an anti-vandalism tool for wikipedia using spatio-temporal analysis of revision metadata. In *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*, page 32. ACM, 2010.

[12] A. B. Yardim, V. Kristof, L. Maystre, and M. Grossglauser. Can who-edits-what predict edit survival? In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2604–2613. ACM, 2018.

[13] W. Yin, H. Schütze, B. Xiang, and B. Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272, 2016.