Enhancing Automatic Readability Assessment with Pre-training and Soft Labels for Ordinal Regression

Jinshan Zeng[1], **Yudong Xie**[1], **Xianglong Yu**[1], **John S. Y. Lee**[2], **Ding-Xuan Zhou**[3]
[1]School of Computer and Information Science, Jiangxi Normal University
[2]Department of Linguistics and Translation, City University of Hong Kong
[3]School of Mathematics and Statistics, University of Sydney
jinshanzeng@jxnu.edu.cn, yudong.xie96@gmail.com,
xianglongyu@jxnu.edu.cn, jsylee@cityu.edu.hk,
dingxuan.zhou@sydney.edu.au

Abstract

The readability assessment task aims to assign a difficulty grade to a text. While neural models have recently demonstrated impressive performance, most do not exploit the ordinal nature of the difficulty grades, and make little effort for model initialization to facilitate fine-tuning. We address these limitations with soft labels for ordinal regression, and with model pre-training through prediction of pairwise relative text difficulty. We incorporate these two components into a model based on hierarchical attention networks, and evaluate its performance on both English and Chinese datasets. Experimental results show that our proposed model outperforms competitive neural models and statistical classifiers on most datasets.

1 Introduction

Readability assessment quantifies the difficulty of a text, that is, the degree to which it can be easily read and understood (McLaughlin, 1969; Klare, 2000). Since an automatic readability assessment (ARA) system can assign a text to a difficulty grade, it is useful for identifying texts or books that are suitable for individuals according to their language proficiency, intellectual and psychological development. ARA research harks back to the last century (Lively and Pressey, 1923; Klare, 1963) and has attracted rising attention in recent years, with impressive performance achieved by many neural approaches (Azpiazu and Pera, 2019; Tseng et al., 2019; Schicchi et al., 2020; Azpiazu and Pera, 2020; Deutsch et al., 2020; Martinc et al., 2021; Lee et al., 2021; Vajjala, 2021; Tanaka-Ishii et al., 2010; Lee and Vajjala, 2022).

There are however a number of limitations in the design and training of current ARA models. First, even though difficulty grades are clearly ordinal in nature, most systems approach the task as multi-class classification with independent labels. During training, texts in adjacent grades (e.g., Grades 2 and 3) are not treated as more similar than those in distant grades (e.g., Grades 2 and 6). Second, although a good initialization can optimize performance in many natural language processing tasks (Tamborrino et al., 2020), state-of-the-art ARA systems generally rely on random initialization (Azpiazu and Pera, 2019; Martinc et al., 2021).

This paper aims to further improve ARA performance by investigating the following research questions:

Ordinal information Can the use of soft labels for ordinal regression (Diaz and Marathe, 2019) improve performance?

Model initialization Can the model be better initialized through pre–training on pairwise relative prediction of text difficulty?

In contrast to most previous work, we conduct both within-corpus and cross-corpus experiments to answer these questions. Within-corpus evaluation may not accurately reflect ARA performance when the model is deployed on texts from other collections. Further, some features (e.g., text length, topics) could be domain-dependent and may not provide the same performance boost in other domains.

The rest of the paper is organized as follows. Following a review of previous work (Section 2), we propose our model (Section 3). We then describe our datasets (Section 4) and the experimental setup (Section 5). Finally, we discuss experimental results (Section 6).

2 Related Work

Early studies in ARA mostly focused on readability formulas, typically developed through empirical pedagogy and psychology (Klare, 1963; Davison and Kantor, 1982). Although these formulas have the advantage of being easily interpretable, they

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4557–4568
December 7-11, 2022

rely on surface features and cannot measure the structure or semantic complexity of a text.

Traditional machine learning methods have been applied to train statistical classifiers for ARA. These classifiers employ a large number of features related to vocabulary, semantics and syntax (Hancke et al., 2012; Sung et al., 2015; Dell'Orletta et al., 2011; Francois and Fairon, 2012; Denning et al., 2016; Arfé et al., 2018; Jiang et al., 2019). Although they often outperform readability formulas, feature engineering and selection can be time-consuming and labor-intensive.

Deep learning methods, which have shown impressive performance in NLP, have recently been applied in ARA. Pre-trained word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) and pre-trained language models such as BERT (Devlin et al., 2019) have been exploited by many neural ARA models (Deutsch et al., 2020; Tseng et al., 2019). *Vec2Read* captures important words and sentences through a multi-level attention mechanism, and uses a Bidirectional Long Short Term Memory (Bi-LSTM) to create representations of whole sentences and individual words. It has performed well on multilingual readability assessment by applying transfer learning (Azpiazu and Pera, 2020).

Most closely related to our model, Hierarchical Attention Networks (HAN) (Yang et al., 2016) consist of both word and sentence encoders to mimic the hierarchical structure of documents. The word encoder uses bidirectional gated recurrent units (Bi-GRU) (Bahdanau et al., 2014) to embed words while summarizing the information from the context. A word-level attention mechanism then aggregates the most informative words to form a sentence vector. At the sentence level, another encoder likewise uses Bi-GRU to embed sentences, and an attention mechanism aggregates the most formative sentences into a text vector. Originally developed for document classification, it has also shown competitive results in ARA (Martinc et al., 2021). Our proposed model follows the architecture of HAN, but uses a pre-trained BERT and a Bi-LSTM instead of Bi-GRUs in the word encoder and sentence encoder, repectively. Further, it adopts soft labels for ordinal regression and a novel pre-training task for initialization.

Combining neural models with hand-crafted linguistic features can further improve performance (Lee et al., 2021). Since our focus is on neural models that do not require feature engineering, we do not pursue this direction of research.

3 Proposed Model

We propose an ARA model based on hierarchical attention networks (HAN) (Yang et al., 2016) with two novel components: the use of soft labels to exploit the ordinal nature of the readability assessment task, and a novel pre-training task for model initialization. We will henceforth refer to this model as *DTRA* (deep text readability assessment).

The proposed model consists of three components (Figure 1). The feature representation component (Section 3.1), similar to HAN, constructs the representation for an input text. A fully connected layer following this component is utilized as a classifier, where the cross entropy loss is adopted as the loss function. The soft-label component (Section 3.2) exploits the ordinal nature of our task by converting discrete grades into soft labels with a distance metric between grades. The pre-training component (Section 3.3) aims to produce a good initialization for fine-tuning.

3.1 Feature Representation Component

The feature representation component produces the text-level representation of an input text. As shown in Figure 1, it consists of the word encoder, word attention, sentence encoder, and sentence attention modules.

3.1.1 Word Encoder Module

The word encoder module uses a pre-trained BERT as the feature extractor. It is stacked by 12-layer Transformer encoder (Vaswani et al., 2017) through residual connection. Its input structure consists of token embedding, segment embedding and position embedding, same as the original BERT (Devlin et al., 2019). Each sentence in the input text is inputted to a BERT, token by token, with all BERTs sharing the same parameters. Let $h_i^t = \mathrm{BERT}(x_i)$ represent the output of the word encoder module for the i-th sentence x_i.

Unlike Vec2Read (Azpiazu and Pera, 2019) and HAN (Yang et al., 2016), which respectively use a Bi-LSTM and a Bi-GRU together with a pre-trained static word embedding in the word encoder module, we use BERT to take advantage of its dynamic word embedding, and also to avoid word segmentation ambiguity for Chinese texts.

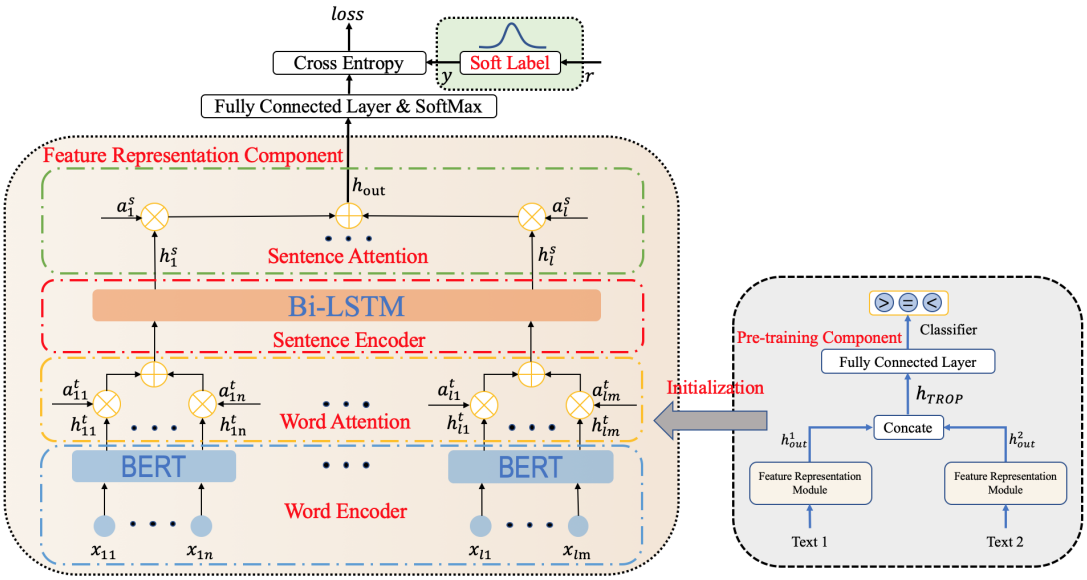


Figure 1: Overview of the proposed model, which consists of the feature representation component (Section 3.1), the soft-label component (Section 3.2), and the pre-training component (Section 3.3)

3.1.2 Word Attention Module

Similar to Vec2Read (Azpiazu and Pera, 2019), we use a token-level attention mechanism to pay more attention to those words of higher significance for the readability assessment of the text. It consists of a single hidden layer neural network to assign the corresponding weight a_{ij}^t to h_{ij}^t, defined as:

$$a_{ij}^t = \frac{\exp(\hat{a}_{ij}^t)}{\sum_k \exp(\hat{a}_{ik}^t)} \quad (1)$$

where $\hat{a}_{ij}^t = \mathrm{ReLU}(W_2^t \,\mathrm{ReLU}(W_1^t h_{ij}^t + b_1^t) + b_2^t)$; W_1^t and W_2^t are the weights of hidden and output layers, respectively; b_1^t and b_2^t are their associated biases vectors; and ReLU is the rectified linear unit activation function defined as $\mathrm{ReLU}(x) = \max\{0, x\}$. Then, we set:

$$\hat{h}_i^s = \sum_k a_{ik}^t h_{ik}^t \quad (2)$$

as the representation of the i-th sentence, for $i = 1, \ldots, l$. We denote $\hat{h}^s := [\hat{h}_1^s, \hat{h}_2^s, ..., \hat{h}_l^s]$.

3.1.3 Sentence Encoder Module

We capture the sentence-order information in the sentence-level representation with a Bi-LSTM, motivated by its sequential nature. Specifically, we sequentially feed the sentence-level representation $\hat{h}^s$ to the Bi-LSTM in the original sentence order of the text to yield a new sentence-level representation $h^s = [h_1^s, h_2^s, ..., h_l^s]$ incorporated with the correct sentence order, i.e., $h^s = \text{Bi-LSTM}(\hat{h}^s)$. By incorporating the context of its neighboring sentences, this enhanced sentence-level representation is intended to improve the readability assessment accuracy, since the sentence order may encode text logic and cohesion that can facilitate readability assessment.

3.1.4 Sentence Attention Module

Similar to Vec2Read (Azpiazu and Pera, 2019), we use a sentence-level attention mechanism to assign an attention weight to each sentence to reflect its importance in the readability assessment of the text. A single hidden layer neural network is used in the sentence-level attention. Specifically, let a_i^s be the attention weight corresponding to h_i^s, defined as:

$$a_i^s = \frac{\exp(\hat{a}_i^s)}{\sum_k \exp(\hat{a}_k^s)}, i \in \{1, 2, ..., l\} \quad (3)$$

where $\hat{a}_i^s = \mathrm{ReLU}(W_2^s \,\mathrm{ReLU}(W_1^s h_i^s + b_1^s) + b_2^s)$ and W_1^s, b_1^s, W_2^s and b_2^s are weights and bias vectors of the hidden and output layers respectively. The final text-level representation h_{out} of a text for classification is thus $h_{out} = \sum_{k=1}^{l} h_k^s a_k^s$. Following the text module, a fully connected layer serves as the classifier, trained on cross entropy loss as the loss function.

3.2 Soft Labels for Ordinal Regression

ARA is an ordinal classification task since the labels have an underlying order from easy to difficult (e.g., Grade 1 to Grade 12). The severity of a classification error therefore depends on the distance

between the gold and predicted labels. During training, there should be a greater penalty for predicting a Grade 3 text as Grade 6 (a distance of three), for example, than as Grade 2 (a distance of one). Since most existing models use "hard" labels, however, they interpret all wrong classes to be infinitely far away from the true class.

We use *soft* labels (Diaz and Marathe, 2019) to exploit the ordinal nature of the readability assessment task. Given an ordinal classification task with K categories, the *soft* label can be defined as follows:

$$y_i = \frac{\exp\left(-\phi(r_i, r_t)\right)}{\sum_{k=1}^{K} \exp\left(-\phi(r_k, r_t)\right)}, \tag{4}$$

where $r_i \in \mathcal{Y} = \{r_1, r_2, ..., r_K\}$ is the i-th category, r_t is the true category and $\phi(r_i, r_t)$ is a distance metric between two categories.

The boundary between adjacent grades tends to be vague. For example, a Grade 3 text may not be clearly more difficult than a Grade 2 text or easier than a Grade 4 text. However, it should be more difficult than those at grades farther away. We therefore take the distance metric $\phi(r_i, r_t)$ in (4) as the following piece-wise constant function:

$$\phi(r_i, r_t) = \begin{cases} 0, & i = t \\ c, & |i - t| = 1, \\ +\infty, & otherwise \end{cases} \tag{5}$$

where c is a positive hyper-parameter that represents the distance between the true label and its adjacent labels. During training, we convert original hard labels into soft labels according to (4) and (5). We empirically set c to be 1.2 according to experimental results in Appendix B.

3.3 Pre-Training Component

A good initialization is crucial for a neural language model given the highly nonconvex nature of the training loss (Tamborrino et al., 2020). To this end, we propose a pre-training task based on the prediction of pairwise relative difficulty of texts: given any two texts, the task is to predict whether the first has a higher, lower, or the same level of readability as the other. We hypothesize that accurate performance in this related task would yield a good initialization of parameters for the fine-tuning stage of an ARA model. We will refer to this pre-training task as *Text Readability Order Prediction* (TROP).

As shown in Figure 1, we randomly selected two texts from the training set and used the feature representation component of the proposed model to construct their representations h^1_{out} and h^2_{out}. We then feed their concatenation $h_{TROP} = [h^1_{out}; h^2_{out}]$ into a three-way classifier, using cross entropy loss as the training objective.

Grade	CMT		CMER	
	# texts	Text length	# texts	Text length
1	235	108.95	218	145.53
2	320	198.58	217	308.44
3	386	329.48	234	538.35
4	321	425.39	229	628.08
5	282	569.82	200	682.41
6	252	660.89	255	701.29
7	199	1202.13	221	1227.19
8	142	1176.94	205	1324.25
9	134	1443.84	188	1302.54
10	140	1617.08	100	2182.08
11	89	1900.85	96	2252.34
12	121	1930.74	97	2043.69

Table 1: Number of texts and their average length at each grade in the CMT and CMER corpora

4 Data

Our evaluation makes use of five datasets in English and Chinese. We used the 8:1:1 ratio for training, development and test data on all datasets.

Newsela The Newsela corpus[1] contains 10,786 texts distributed among levels 2-12 for English and Spanish. Similar to (Martinc et al., 2021), we removed the documents for Spanish while only focused on the readability assessment for those English documents. Thus, the total number of samples of Newsela used in our experiments is 9,565.

OneStopEnglish The OneStopEnglish corpus[2] was created for English as a second language learners. It contains a total of 567 English texts, with each text written in three versions: elementary, intermediate and advanced.

WeeBit The WeeBit corpus consists of 6,388 English texts from WeeklyReader[3] and BBC-Bitesize[4] in five grades. For a balanced dataset, we randomly sample 625 texts in each grade.

CMT China Mainland Textbook (CMT) (Cheng et al., 2020) consists of a total of 2,723,430

[1]https://newsela.com
[2]https://zenodo.org/record/1219041
[3]http://www.weeklyreader.com
[4]http://www.bbc.co.uk/bitesize

characters, distributed in 2,621 texts in twelve grades, all taken from Chinese textbooks from the first grade of primary school to the third grade of high school in mainland China. Table 1 reports detailed statistics on CMT.

CMER China Mainland Extracurricular Reading (CMER) is a new dataset collected by the authors. It consists of texts from extracurricular reading books for kids and teenagers at China mainland currently on the book market, with a total of 3,395,923 characters, distributed in 2,260 texts in 12 levels. Table 1 reports detailed statistics on CMER.[5]

The two Chinese datasets facilitate cross-corpus evaluation since they follow the same grade scale defined by the national standard, but the materials are compiled independently from different sources. Cross-corpus evaluation would be difficult for the English datasets because of the lack of direct correspondence between their scales.

5 Experimental Set-up

This section presents the baselines to which we will compare our proposed model, the evaluation metrics, and implementation details.

5.1 Neural model baselines

Vec2Read (Azpiazu and Pera, 2019) uses pre-trained static word embedding, a Bi-LSTM, word- and sentence-level attention mechanisms. The embedding size and hidden layer size of Bi-LSTM were set to be 300 and 128 respectively. When adapted to Chinese corpora, we used the model called *FastText* (Bojanowski et al., 2017) to yield the pre-trained word embedding.

BERT (Devlin et al., 2019) uses the default BERT model for fine-tuning and the default learning rate (2e-5).

ALBERT (Lan et al., 2019) uses the factorized embedding parameterization and cross-layer parameter sharing to reduce the size of model (that is, reducing from 108 M to 12 M).

Longformer (Beltagy et al., 2020) uses a variant of self-attention mechanism that scales linearly with sequence length to process long texts.

HAN (Martinc et al., 2021) uses two Bi-LSTMs, word- and sentence-level attention mechanisms to encode word and sentence representations. We used the same settings as Martinc et al. (2021), where word and sentence embedding sizes were 200 and 100 respectively.

Lite-DTRA To reduce the requirement of storage memory of the hardware, we provide a lite version of the proposed model, where the pre-trained BERT with frozen parameters is replaced by a lite version of BERT, i.e., ALBERT (Lan et al., 2019), and thus allows the model to be trained in an end-to-end way.

All deep learning models were implemented in Pytorch, Transformers (Wolf et al., 2020), AMD 3900x, GeForce RTX 3090 environment.

5.2 Traditional classifier baselines

We report performance of the traditional machine learning methods on the Chinese datasets (CMT and CMER):

Logistic Regression, Support Vector Machine (SVM), Random Forest, Naive Bayes. As shown in Table 7 (Appendix A), we manually extracted 43 features at the lexical, syntactic, semantic and cohesion levels, mainly taken from Sung et al. (2015). These traditional machine learning methods are not evaluated on the English datasets since their performance has already been extensively reported in previous research (Martinc et al., 2021; Lee et al., 2021).

The hyperparameters were tuned on development data via 10-fold cross-validation. All methods were implemented in Matlab R2017b, Intel(R) Xeon(R) E5-2667 environment.

5.3 Evaluation metrics

Our evaluation metrics include *classification accuracy* (C-acc), *adjacent accuracy* (A-acc) and the macro *F1-measure* (F1). Adjacent accuracy is defined as the proportion of samples with the predicted labels adjacent to the gold labels (Sung et al., 2015), motivated by the strong ambiguity between the adjacent classes.

5.4 Implementation details

For **DTRA**, we used a pre-trained BERT[6] for texts in the word encoder module, where the number of Transformer encoder layers is 12 and the output feature size is 768. The sizes of the input, hidden and output layers in the token-level attention mechanism are 768, 192 and 1, respectively. In particular, we froze parameters of BERT in DTRA

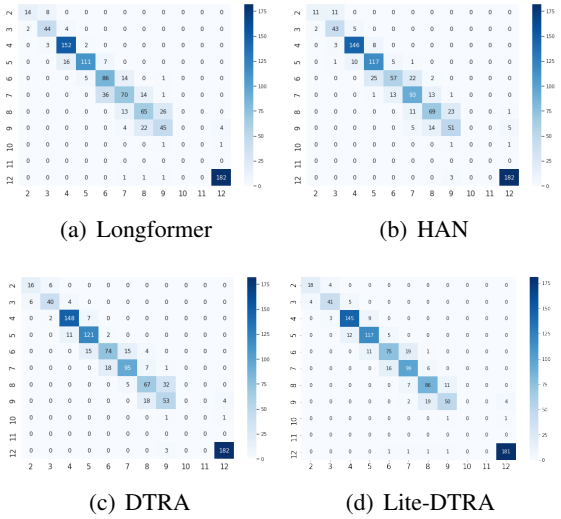
[5]This dataset is accessible at https://github.com/JinshanZeng/DTRA-Readability

[6]For English: https://huggingface.co/bert-base-uncased; For Chinese: https://huggingface.co/bert-base-chinese

due to the limitation of storage memory of the hardware. The sizes of the input and hidden layers in the Bi-LSTM are 768 and 256, respectively, and the sizes of the input, hidden and output layers in the sentence-level attention are 512, 128 and 1, respectively. Following the sentence attention module, there is a fully connected layer as the classifier. We used the cross-entropy loss and Adam algorithm (Kingma and Ba, 2015) as the optimizer to fine tune the proposed model. A weighting decay regularization with the regularization parameter 0.01 was also adopted. In the pre-training component, the initial learning rates of Adam for the training of these three modules and fully connected layer were all 7e-5, while in the fine-tuning stage, they were set to be 1e-5 and 4e-5 respectively. **Lite-DTRA** follows the same settings as DTRA, except that the frozen BERT is replaced with ALBERT (Lan et al., 2019).

6 Experimental Results

We report experimental results in English datasets (Section 6.1) and Chinese datasets (Section 6.2). We then present an ablation study (Section 6.3) and a comparison between the soft labels and regression (Section 6.4). Finally, we discuss results of a cross-corpus evaluation with few-shot learning (Section 6.5).



(a) Longformer (b) HAN

(c) DTRA (d) Lite-DTRA

Figure 2: Confusion matrices of four deep learning models over Newsela. The horizontal- and vertical-axis of each figure represent the predicted categories and the true categories of the samples, respectively.

6.1 English datasets

As shown in Table 2, DTRA achieved higher accuracy on Newsela (83.26%) and OneStopEnglish (85.00%) than all five baseline neural models. These compare favorably with the best result achieved by HAN on Newsela (81.38%) and OneStopEnglish (78.72%) reported by Martinc et al. (2021) and by BERT on OneStopEnglish reported by Lee et al.(2021).[7] On WeeBit, however, BERT achieved the highest accuracy, which is similar to the result reported by Martinc et al. (2021). Evaluation on F1-measure exhibits the same trend. These experimental results suggest the effectiveness of the soft labels and pre-training. The individual contribution of these components will be further analyzed in the ablation study.

In all settings, Lite-DTRA offered slightly better performance than DTRA. This may be attributable to the end-to-end training with ALBERT, in contrast to the pre-trained BERT's frozen parameters for feature extraction.

To visualize the performance of proposed models, Figure 2 shows the confusion matrices of the top-four models (Longformer, HAN, DTRA and Lite-DTRA) in terms of accuracy on Newsela. The values in confusion matrices of DTRA and Lite-DTRA are more concentrated on the diagonal than other two models, in particular at the seventh and ninth grades.

6.2 Chinese datasets

As shown in Table 3, DTRA achieved 44.42% accuracy on CMT and 26.50% on CMER. The lower accuracy in comparison to the English results is expected since CMT and CMER contain 2,000 texts approximately but have 12 grades. On both accuracy and F1-measure, DTRA outperformed all four neural baseline models as well as all statistical classifiers. On CMT, HAN achieved the second highest accuracy (42.53%), while the LR classifier performed second best (24.98%) on CMER.

6.3 Ablation Studies

We conducted an ablation study to measure the contribution of the soft labels and the pre-training to DTRA's performance. The top of Table 4 compares the performance of the complete DTRA and its performance upon removal of the pre-training

[7]The results are not directly comparable since we were not able to obtain the dataset splits used in Martinc et al. (2021) and Lee et al. (2021)

Model		Vec2Read	BERT	ALBERT	Longformer	HAN	**DTRA**	**Lite-DTRA**
Newsela	C-acc	47.18	77.09	78.77	80.44	80.44	83.26	**84.94**
	A-acc	64.33	98.54	97.80	98.54	97.80	**98.85**	98.75
	F1	24.83	76.80	77.94	80.17	79.95	83.11	**84.74**
OneStop English	C-acc	43.33	80.00	83.33	81.67	78.33	85.00	**86.67**
	A-acc	63.33	**100**	**100**	**100**	**100**	**100**	**100**
	F1	44.91	79.77	82.71	81.68	78.61	84.91	**86.79**
WeeBit	C-acc	75.82	**87.74**	86.77	85.48	80.97	84.84	85.48
	A-acc	96.13	98.07	99.36	99.03	97.42	99.03	**100**
	F1	75.31	**87.85**	86.80	85.53	80.92	84.84	85.45

Table 2: ARA performance on English datasets in terms of accuracy (C-acc), adjacent accuracy (A-acc) and F1, in percentage. The best and second best results are marked in bold and blue color, respectively.

Model		LR	SVM	RF	Bayes	Vec2Read	BERT	ALBERT	HAN	**DTRA**	**Lite-DTRA**
CMT	C-acc	32.25	35.85	39.12	33.49	34.59	33.84	36.30	42.53	**44.42**	**44.42**
	A-acc	68.55	73.22	74.97	70.25	68.05	67.11	70.51	79.58	81.10	**82.04**
	F1	30.65	35.39	37.87	30.82	28.58	31.35	33.26	41.09	**43.87**	42.87
CMER	C-acc	24.98	24.51	21.57	20.38	24.95	22.74	22.96	23.40	**26.50**	**26.50**
	A-acc	53.00	54.98	52.41	46.22	53.86	47.68	49.89	54.53	58.50	**62.47**
	F1	23.25	24.06	20.31	16.37	20.57	17.16	22.06	18.48	**25.16**	22.06

Table 3: ARA performance on the Chinese datasets in terms of accuracy (C-acc), adjacent accuracy (A-acc) and F1, in precentages. The best and second best results are marked in bold and blue color, respectively.

step and soft labels for ordinal regression. On most datasets, there was a decrease in both accuracy and F1 after removal of pre-training, indicating its utility for ARA. The use of soft label improved the accuracy on all datasets except OneStopEnglish. In terms of F1-measure, the soft labels were helpful on Newsela and Weebit but slightly hurt performance on OneStopEnglish, CMT and CMER.

The bottom of Table 4 compares Lite-DTRA and its counterpart version with frozen ALBERT parameters (referred to as **Lite-DTRA-frozen**). Lite-DTRA obtained better results on all metrics for all datasets, except F1 for CMER. This suggests that using the trainable ALBERT model is beneficial for for Lite-DTRA in practice.

6.4 Soft labels vs. regression

ARA can be formulated as a regression, classification or ordinal regression task. We further examined the effect of the soft labels through a comparison with standard regression and multi-class classification.

The version of DTRA without pre-training, which will be referred to as the *Ordinal-DTRA* model, serves as the reference point. We directly used the features output by the feature representation component to train a classification model, which will be called the *Classification-DTRA* model. We used the same features to train a regression model, which will be called the *Regression-DTRA* model.

Table 5 compares the accuracy of these three models. Ordinal-DTRA gave the best performance over all five datasets. Classification-DTRA achieved the second best performance, and outperformed Regression-DTRA with a substantial gap in most datasets. These results suggest that the soft labels are more effective in capturing the ordinal nature of the readability grades than direct use of multi-class classification or regression.

6.5 Cross-corpus evaluation

Since ARA models may be used in predicting the difficulty of texts from other sources, we gauge the robustness of our proposed model in a cross-corpus evaluation. We conducted experiments in two settings: (a) train DTRA on CMER, and test on CMT; and (b) train DTRA on CMT, and test on CMER. We did not attempt cross-corpus evaluation in English because of the lack of direct mapping among the readability scales adopted in Newsela, OneStopEnglish and WeeBit.

In each setting, we further evaluated the impact of limited quantities of samples in the target corpus for few-shot learning. Specifically, we evaluated model performance when {0, 5, 10, 15, 20, 25, 30} samples from the target corpus were added to the training data. As shown in Table 6, when trained only on CMER, DTRA performed at 31.00% on CMT, which constitutes a 13% degradation compared to the within-corpus setting (44.42%). It outperformed all other baselines both on accuracy and F1. On CMER, DTRA performed at 19.87% on CMER, a 7% degration compared to the within-corpus setting (26.5%). While it outperforms all other baselines on F1, Vec2Read achieved the highest accuracy (21.95%). These results suggest that our models not only captured characteristics pe-

Model	Newsela			OneStopEnglish			WeeBit			CMT			CMER		
	C-acc	A-acc	F1	C-acc	A-acc	F1	C-acc	A-acc	F1	C-acc	A-acc	F1	C-acc	A-acc	F1
DTRA	**83.26**	**98.85**	**83.11**	**85.00**	**100**	**84.91**	**84.84**	99.03	**84.84**	**44.42**	**81.10**	**43.87**	**26.50**	**58.50**	**25.16**
w/o pre-training	81.59	98.75	81.18	81.67	**100**	81.21	83.23	**99.36**	82.83	44.05	80.15	41.04	26.05	56.73	25.04
w/o soft labels	80.96	98.54	80.66	81.67	**100**	81.26	82.26	98.71	82.30	43.48	78.83	41.33	25.83	56.51	25.12
Lite-DTRA	**84.94**	**98.75**	**84.74**	**86.67**	**100**	**86.79**	**85.48**	**100**	**85.45**	**44.42**	**82.04**	**42.87**	**26.50**	**62.47**	22.06
Lite-DTRA-frozen	81.38	**98.75**	81.06	80.00	**100**	79.97	84.19	99.36	84.13	43.67	79.77	40.07	25.83	60.71	**23.33**

Table 4: Ablation study on DTRA (top) and the impact of frozen ALBERT parameters (bottom). The best results are bolded.

Model	Regression-DTRA	Classification-DTRA	Ordinal-DTRA
Newsela	80.96	80.96	**81.59**
OneStop English	73.68	**81.67**	**81.67**
WeeBit	78.71	82.26	**83.23**
CMT	32.97	43.48	**44.05**
CMER	17.44	25.83	**26.05**

Table 5: The accuracy of DTRA based on regression, classification and ordinal regression, in percentages. The best results are marked in bold.

culiar to textbooks (Section 6.1), but also learned textual difficulty features that can be effectively transferred to other texts.

Even a small amount of data from the target corpus could improve model performance. For example, just five texts from the training set of the target corpus per grade could already boost the accuracy of DTRA by about 5% absolute (36.11% on CMT and 25.17% on CMER). It also outperformed all other baselines in both accuracy and F1-measure. The amount of training samples from the target corpus is mostly positively correlated with model performance. Lite-DTRA tends to perform better than DTRA in leveraging limited data. With 15 samples, it achieved 41.59% accuracy on CMT, which is within 3% of its accuracy in the within-corpus setting; on CMER, it surpassed its within-corpus performance with an accuracy of 28.26%, which may indicate the relatively high quality of the CMT training data.

7 Conclusion

This paper has proposed a deep learning model for text readability assessment that achieved competitive performance on the benchmark datasets Newsela, OneStopEnglish and WeeBit. Our model, which is based on Hierarchical Attention Network (HAN) (Yang et al., 2016), incorporates two novel elements: soft labels for ordinal regression (Diaz and Marathe, 2019), and a pre-training task on pairwise relative text difficulty that aims to improve the model initialization for fine-tuning.

We conducted experiments on both English and Chinese datasets to compare this model to a number of competitive neural models, including BERT, Vec2Read and HAN. The proposed model outperforms all baselines on most datasets in terms of both accuracy and F1. A lite version of the proposed model, with reduced storage memory requirement, also offered competitive performance. An ablation study demonstrated that the pre-training and the soft labels brought benefits in most datasets. The proposed model also outperformed most baselines in the cross-corpus setting, demonstrating its ability to learn features of text difficulty that are transferable to other kinds of texts.

Samples from target corpus	Training data	CMER			CMT		
	Test data	CMT			CMER		
	Model	C-acc	A-acc	F1	C-acc	A-acc	F1
0	Vec2Read	20.23	56.90	15.05	**21.85**	55.19	18.38
	BERT	20.98	53.69	16.64	20.09	53.20	19.30
	HAN	27.22	58.41	20.84	19.34	**55.41**	18.94
	DTRA	**31.00**	65.97	**29.78**	19.87	51.88	**19.78**
	Lite-DTRA	29.87	**66.16**	25.13	18.76	52.10	17.87
5	Vec2Read	29.12	61.91	24.76	22.96	**57.84**	16.70
	BERT	27.60	62.95	25.05	21.19	52.10	18.78
	HAN	32.23	70.51	26.62	23.62	54.31	21.73
	DTRA	**36.11**	71.83	**34.48**	25.17	56.73	**24.00**
	Lite-DTRA	**36.11**	**75.24**	33.52	**26.27**	55.19	23.42
10	Vec2Read	30.95	64.84	26.99	22.30	53.20	19.97
	BERT	29.68	58.24	22.31	22.96	51.88	19.77
	HAN	34.43	70.33	27.34	22.96	**58.72**	21.16
	DTRA	34.62	71.80	30.05	24.28	58.06	**21.67**
	Lite-DTRA	**34.97**	**75.24**	**31.29**	**24.50**	57.62	19.28
15	Vec2Read	28.94	64.65	26.17	23.18	55.85	19.72
	BERT	29.68	62.76	26.27	22.74	49.67	21.91
	HAN	37.55	74.91	33.00	24.72	**62.25**	21.93
	DTRA	36.48	76.37	36.37	24.95	58.50	23.14
	Lite-DTRA	**41.59**	**79.01**	**38.42**	**28.26**	58.50	**27.38**
20	Vec2Read	30.22	61.91	25.87	24.06	**56.29**	20.68
	BERT	29.85	61.36	20.19	24.28	50.99	23.17
	HAN	35.99	72.53	29.83	24.50	**56.29**	22.19
	DTRA	36.67	74.67	35.74	24.72	**56.29**	22.80
	Lite-DTRA	**39.89**	**76.56**	**37.99**	**25.17**	56.07	**24.07**
25	Vec2Read	30.95	69.60	26.20	24.06	56.95	19.39
	BERT	30.25	56.33	26.79	23.18	48.12	22.39
	HAN	37.73	**78.02**	33.59	26.93	**61.37**	23.52
	DTRA	35.92	75.80	34.19	26.40	56.95	25.72
	Lite-DTRA	**38.75**	76.18	**35.61**	**27.81**	59.16	**25.80**
30	Vec2Read	31.32	68.50	28.51	22.08	53.64	17.42
	BERT	30.25	63.14	28.01	23.62	48.12	23.24
	HAN	**40.66**	78.02	34.48	25.61	**59.60**	23.98
	DTRA	38.37	73.91	**37.86**	**27.59**	58.94	**25.93**
	Lite-DTRA	40.08	**79.02**	36.48	25.61	56.29	24.08

Table 6: ARA performance in cross-corpus evaluation. The best and second best results are marked in bold and blue color, respectively.

8 Limitations

This research has not considered hybrid models, which combine manual linguistic features with the neural networks. Accuracy in the cross-corpus set-

ting could also be further improved with transfer learning techniques. There is still much room for improvement in the performance of the ARA model before it is ready for deployment in the classroom for automatic assignment of reading materials to students.

Acknowledgements

J. Zeng was partly supported by National Natural Science Foundation of China (No. 61977038) and the Thousand Talents Plan of Jiangxi Province (No. jxsq2019201124). J. Lee was partly supported by the Language Fund from the Standing Committee on Language Education and Research (project EDB(LE)/P&R/EL/203/14) and by the General Research Fund (project 11207320). This paper in its first version was written when D.-X. Zhou worked at City University of Hong Kong, supported partially by the Research Grants Council of Hong Kong [Project Numbers CityU 11308020, N_CityU102/20, C1013-21GF], Laboratory for AI-powered Financial Technologies, Hong Kong Institute for Data Science, Germany/Hong Kong Joint Research Scheme [Project No. G-CityU101/20], and National Natural Science Foundation of China [Project No. 12061160462].

References

Barbara Arfé, Lucia Mason, and Inmaculada Fajardo. 2018. Simplifying informational text structure for struggling readers. *Reading and Writing*, 31(9):2191–2210.

Ion Madrazo Azpiazu and Maria Soledad Pera. 2019. Multiattentive recurrent neural network architecture for multilingual readability assessment. *Transactions of the Association for Computational Linguistics*, 7:421–436.

Ion Madrazo Azpiazu and Maria Soledad Pera. 2020. An analysis of transfer learning methods for multilingual readability assessment. In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pages 95–100.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Yong Cheng, Dekuan Xu, and Jun Dong. 2020. On key factors of text reading difficulty grading and readability formula based on chinese textbook corpus [in chinese]. *Applied Linguistics*, 1:132–143.

Alice Davison and Robert N. Kantor. 1982. On the failure of readability formulas to define readable texts: A case study from adaptations. *Reading Research Quarterly*, 17(2):187–209.

Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi. 2011. Read–it: Assessing readability of italian texts with a view to text simplification. *Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies*, pages 73–83.

Joel Denning, Maria Soledad Pera, and Yiu-Kai Ng. 2016. A readability level prediction tool for k-12 books. *association for information science and technology*, 67(3):550–565.

Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic features for readability assessment. In *Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Raul Diaz and Amit Marathe. 2019. Soft labels for ordinal regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4738–4747.

Thomas Francois and Cedrick Fairon. 2012. An "ai readability" formula for french as a foreign language. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 466–477.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability classification for german using lexical, syntactic, and morphological features. In *Proceedings of COLING 2012: Technical Papers*, page 1063–1080.

Zhiwei Jiang, Qing Gu, Yafeng Yin, Jianxiang Wang, and Daoxu Chen. 2019. Graw+: A two-view graph propagation method with word coupling for readability assessment. *Journal of the Association for Information Science and Technology*, 70(5):433–447.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

George R. Klare. 1963. *Measurement of readability*. Iowa State University Press, Iowa, USA.

George R. Klare. 2000. The measurement of readability: useful information for communicators. *ACM Journal of Computer Documentation*, 24(3):107–121.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Bruce W. Lee, Yoo Sung Jang, and Jason Hyung-Jong Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1069–1086.

Justin Lee and Sowmya Vajjala. 2022. A neural pairwise ranking model for readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3802–3813, Dublin, Ireland. Association for Computational Linguistics.

Bertha A Lively and Sidney L Pressey. 1923. A method for measuring the vocabulary burden of textbooks. *Educational administration and supervision*, 9(7):389–398.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Computational Linguistics*, 47(1):141–179.

G. Harry McLaughlin. 1969. Smog grading - a new readability formula. *The Journal of Reading*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Daniele Schicchi, Giovanni Pilato, and Giosué Lo Bosco. 2020. Deep neural attention-based model for the evaluation of italian sentences complexity. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pages 253–256. IEEE.

Yao-Ting Sung, Wei-Chun Lin, Scott Benjamin Dyson, Kuo-En Chang, and Yu-Chia Chen. 2015. Leveling l2 texts through readability: Combining multilevel linguistic features with the cefr. *The Modern Language Journal*, 99(2):371–391.

Alexandre Tamborrino, Nicola Pellicano, Baptiste Pannier, Pascal Voitot, and Louise Naudin. 2020. Pretraining is (almost) all you need: An application to commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 3878–3887.

Kumiko Tanaka-Ishii, Satoshi Tezuka, and Hiroshi Terada. 2010. Sorting texts by readability. *Computational linguistics*, 36(2):203–227.

Hou-Chiang Tseng, Hsueh-Chih Chen, Kuo-En Chang, Yao-Ting Sung, and Berlin Chen. 2019. An innovative bert-based readability model. In *International Conference on Innovative Technologies and Learning (ICITL) 2019: Innovative Technologies and Learning*, pages 301–308.

Sowmya Vajjala. 2021. Trends, limitations and open challenges in automatic readability assessment research. *arXiv:2105.00973*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Appendix A. Linguistics Features

Table 7 lists the 43 features, based on those proposed by Sung et al. (2015), that were used for training the Linear Regression (LR), Support Vector Machine (SVM), Random Forest (RF) and Naive Bayes (NB) classifiers (Section 5.2).

Appendix B. Hyperparameter Tuning

The hyperparameter c for the soft labels (Section 3.2) was tuned on the development set of the Newsela corpus. As shown in Figure 3, the performance of DTRA was optimal when $c = 1.2$. We thus set c to be 1.2 in our experiments.

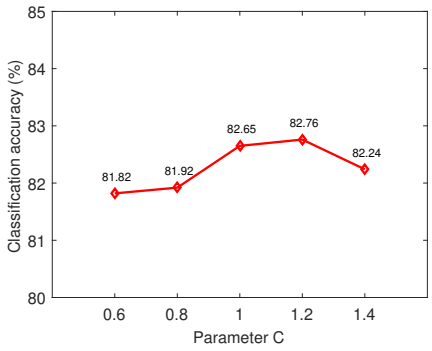


Figure 3: Classification accuracy of DTRA over different values of the hyperparameter c for the soft labels

Category	Feature name	Definition
Lexical level	1: Characters	Total number of characters
	2: Words	Total number of words
	3: Adverbs	Total number of adverbs
	4: Verbs	Total number of verbs
	5: Low stroke-count characters	Total number of characters with 1-7 strokes
	6: Intermediate stroke-count characters (8˜15 strokes)	Total number of characters with 8-15 strokes
	7: High stroke–count characters (>15)	Total number of characters with more than 15 strokes
	8: Ratio of Low stroke-count characters	Proportion of ow stroke-count characters
	9: Ratio of Intermediate stroke-count characters (8˜15 strokes)	Proportion of Intermediate stroke-count characters
	10: Ratio of High stroke-count characters	Proportion of High stroke-count characters
	11: Average strokes	Total number of strokes of each character divided by the number of characters
	12: Two-character words	Total number of two-character words
	13: Three-character words	Total number of three-character words
	14: level 0 words	Total number of words not in 8,000 Chinese Words
	15: level 0 words ratio	level 0 words divided by the total number of words
	16-22: level 1,2,...,7 words	Total number of words in level 1,2,...,7 respectively
	23-29: level 1,2,...,7 words ratio	level 1,2,...,7 divided by the total number of words respectively
	30: Average of vocabulary levels in 8k	Total word difficulty, as according to 8,000 Chinese Words, divided by the total number of words in 8,000 Chinese Words
	31: Average of vocabulary levels	Total word difficulty, as according to 8,000 Chinese Words, divided by the total number of words
	32: Mean square of vocabulary levels	Sum of the squares of word difficulty, as defined by 8,000 Chinese Words, divided by the total number of words
	33: Mean square of vocabulary levels in 8k	Sum of the squares of word difficulty, as defined by 8,000 Chinese Words, divided by the total number of words in 8,000 Chinese Words
	34: High-level words	Sum of words belonging to the vantage and effective operational proficiency levels of 8,000 Chinese Words
	35: High-level words ratio	High-level words divided by the total number of words in 8,000 Chinese Words
Semantic Level	36: Content words	Total number of content words
	37: Frequency of content words	Frequency of content words
Syntactic level	38: Average sentence length	Total number of words divided by the total number of sentences
Cohesion Level	39: Pronouns	Total number of pronouns
	40: Conjunctions	Total number of conjunctions
	41: Personal pronouns	Total number of personal pronouns
	42: First person pronouns	Total number of first person pronouns
	43: Third person pronouns	Total number of third person pronouns

Table 7: Features used for training the LR, SVM, RF and NB classifiers