Evaluation of Unsupervised Automatic Readability Assessors Using Rank Correlations

Yo Ehara
Tokyo Gakugei University
4-1-1 Nukuikita-machi, Koganei-shi, Tokyo 184-8501 Japan
ehara@u-gakugei.ac.jp

Abstract

Automatic readability assessment (ARA) is the task of automatically assessing readability with little or no human supervision. ARA is essential for many second language acquisition applications to reduce the workload of annotators, who are usually language teachers. Previous unsupervised approaches manually searched textual features that correlated well with readability labels, such as perplexity scores of large language models. This paper argues that, to evaluate an assessors' performance, rank-correlation coefficients should be used instead of Pearson's correlation coefficient (ρ). In the experiments, we show that its performance can be easily underestimated using Pearson's ρ, which is significantly affected by the linearity of the output readability scores. We also propose a lightweight unsupervised readability assessor that achieved the best performance in both the rank correlations and Pearson's ρ among all unsupervised assessors compared.

1 Introduction

Assessing readability plays an essential role in second language acquisition; it can be used for many educational applications such as intelligent reading support systems and placement tests for language classes. Readability assessment is a costly task for educational experts and language teachers. To perform it, they must read a text and assess its readability by guessing how difficult the text is for target learning readers. Hence, to reduce the cost of the labor required by educational experts, the task of automatically identifying the readability of texts for language learners, known as automatic readability assessment (ARA), has been extensively studied in the field of artificial intelligence (AI).

Unsupervised automatic readability assessment appeared early but has recently been reexamined as a research focus. Early studies such as the Dale-Chall formula (1948) (Dale and Chall, 1948), the Flesch Reading Ease formula (Flesch, 1948) (1948), and the Flesch-Kincaid readability formula (1975) (Kincaid et al., 1975) were *unsupervised*, as they did not use costly annotated readability labels. Given a text, these formulae calculate its readability score based on simple superficial textual features such as the average length of a word in the given text. However, most of these early formulae are designed to assess readability for children who are native speakers. Evaluation datasets with readability labels annotated by language teachers targeting second language learners appeared much later, in the 2010s (Feng et al., 2010; Xia et al., 2016; Vajjala and Lučić, 2018). In these works, automatic readability assessment tasks using these evaluation datasets were formalized as a supervised document classification problem, and substantial research efforts were invested into the construction of classifiers by feature engineering to find complicated textual features that correlate well with readability labels.

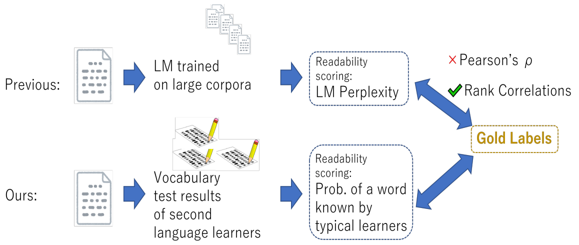
Previous:
LM trained on large corpora
Readability scoring: LM Perplexity
× Pearson's ρ
✓ Rank Correlations
Gold Labels
Ours:
Vocabulary test results of second language learners
Readability scoring: Prob. of a word known by typical learners

Figure 1: Overview of the previous and our approaches.

Recently, Martinc et al. (2021) revisited the unsupervised approach. They proposed that the perplexity scores of neural language models can also be used to represent the readability of text for second language learners and proposed to use them for unsupervised automatic readability assessment. The upper part of Fig. 1 show their approach. Given a text, their method uses no valuable readability label for training but uses only the language model trained on other large corpora, their method pre-

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 62–72
July 5–10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.26615/978-954-452-056-4_007

dicts the text's readability score as an output.

While this idea is sound, however, their evaluation is validated using the correlation coefficient, or Pearson's ρ, as illustrated on the right-hand side of Fig. 1. Pearson's ρ measures the degree of *linear* correlation between two random variables (Mukaka, 2012). As neither the readability levels of the evaluation corpora nor the readability scores output by unsupervised readability assessors are necessarily linear, the use of Pearson's ρ can lead to inaccurate evaluation.

This study investigates how the use of Pearson's ρ affects the evaluation of unsupervised readability assessors. We analyze how unsupervised assessors' performance can be easily underestimated if the readability scores are not linear. For this purpose, we also build a lightweight unsupervised readability assessor, denoted by the lower part of Fig. 1. We show that, alternatively, rank-correlation coefficients are more robust to the linearity and appropriate for this evaluation.

The contributions of this paper are summarized as follows.

- We indicate that the previous evaluation of unsupervised readability assessors by using Pearson's ρ is problematic.
- We demonstrate the degree by which Pearson's ρ underestimates the readability score without linearity on a publicly available reliable evaluation dataset.
- We show that, instead of Pearson's ρ, the use of rank-correlation coefficients is appropriate.
- We propose a novel lightweight unsupervised readability assessor that achieves best performance in terms of both Pearson's ρ and rank-correlation coefficients.

2 Automatic Readability Assessment

This section formalizes the problem of automatic readability assessment. Let us suppose that we have N texts to assess: we write the set of texts as $\{\mathcal{T}_i | i \in \{1, \ldots, N\}\}$. Let $\mathcal{Y}$ be the set of readability labels. Labels are typically ordered in the order of difficulty. For example, in the *OneStopEnglish* dataset (Vajjala and Lučić, 2018), we can set $\mathcal{Y} = \{0, 1, 2\}$, where 0 is elementary, 1 is intermediate, and 2 is advanced. The number of levels depends on the evaluation corpus. Using $\mathcal{Y}$, we write the label for $\mathcal{T}_i$ as $y_i \in \mathcal{Y}$.

2.1 Goal in Unsupervised Setting

Given each text $\mathcal{T}_i$, an *assessor* outputs its readability score s_i. In a supervised setting, the *assessor* knows the number of levels in the evaluation corpus from training examples. Hence, s_i ranges within $\mathcal{Y}$: $s_i \in \mathcal{Y}$. However, in an unsupervised setting, it is noteworthy that the assessor does not know $\mathcal{Y}$, or how many levels the evaluation corpus has, because no label is given. Hence, even if only integers are allowed for y_i, s_i can be a real value.

Throughout this paper, we write arrays using [and]. Given N texts $[\mathcal{T}_i | i \in \{1, \ldots, N\}]$, our goal is to make an assessor output arrays of readability scores $[s_i | i \in \{1, \ldots, N\}]$ that *correlate well* with the array of labels $[y_i | i \in \{1, \ldots, N\}]$. Here, there are multiple types of correlation coefficients between the array of scores and the array of labels, which we explain in the later sections. Typically, we should use *rank coefficients* when s_i is real-valued.

2.2 Evaluation and Correlation coefficients

In most evaluation datasets, educational experts are asked to assess text readability by choosing a label from the set of predefined readability labels, $\mathcal{Y}$. In contrast, automatic readability assessors output real-valued scores in an unsupervised setting. How do we compare readability level labels and real-valued scores?

A simple but naïve way to make this comparison is to use the Pearson correlation coefficient $\rho_{y,s}$, which is defined as follows:

$$\rho_{y,s} = \frac{cov(y, s)}{\sigma_y \sigma_s} \qquad (1)$$

In Eq. 1, $cov(y, s)$ denotes the covariance between y and s, σ_y denotes the standard deviation of y, and σ_s denotes the standard deviation of s. Eq. 1 ranges $[-1, 1]$, where 1 is the perfect correlation.

However, the Pearson correlation coefficient measures the degree of *linear* correlation between two random variables (Mukaka, 2012). The readability levels of evaluation corpora are not necessarily linearly distributed. Readability scores that the assessors output are also not necessarily linear. In these cases, it is usually more appropriate to focus on the correlation between the *rankings* of the readability label y_is and scores s_is. *Rank correlation coefficients* measure the correlation between two rankings with the range of $[-1, 1]$. Two types of them are notable: Spearman's ρ and Kendall's τ (Alvo and Philip, 2014).

Spearman's ρ is defined as the Pearson's ρ between two rankings. We first convert labels into rankings: rg_y, and convert scores into rankings: rg_s. Then, using Eq. 1, Spearman's ρ is defined as $\rho_{\text{rg}_y \text{rg}_s}$. When converting labels into rankings, texts that have the same level are regarded as *ties* in a ranking. While there are many ways to handle ties, the *mid-rank* method is usually used in calculating Spearman's ρ (Amerise and Tarsitano, 2015). This method simply uses the average of ranks for the rank of a tie. For example, let us consider an array of labels $[2, 1, 1, 0]$. The two 1s in this array are ties taking the 2nd and 3rd ranks. As the average of 2 and 3 is 2.5, the *mid-rank* ranking of this array is $[4, 2.5, 2.5, 1]$.

We first introduce the definition of Kendall's τ when there are no ties as follows.

$$\tau = \frac{n_c - n_d}{\text{Num. of all pairs}} \tag{2}$$

Kendall's τ focuses on the pairs of the given arrays: in our setting, (y_i, s_i) and $(y_{i'}, s_{i'})$ where $i < i'$. n_c denotes the number of concordant pairs, n_d denotes the number of discordant pairs. The pair is said to be concordant if either both $y_i < y_{i'}$ and $s_i < s_{i'}$ hold or $y_i > y_{i'}$ and $s_i > s_{i'}$; otherwise, the pair is said to be discordant. If $y_i = y_{i'}$, we call y_i and $y_{i'}$ ties. The same holds for s. Num. of all pairs $= \frac{1}{2}N(N-1)$ when there are no ties.

In reality, y has many ties, so Eq. 2 cannot be used for the evaluation.. There are multiple correction methods to account for ties in Kendall's τ; they are named τ-a, τ-b, and τ-c. In our setting, namely unsupervised readability assessment, tau-c should be used because y and s may have different scales.

τ-b can be described as follows [1].

$$\tau_b = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \tag{3}$$

n_c denotes the number of concordant pairs, n_d denotes the number of discordant pairs. $n_0 = N(N-1)/2$, n_1 is the sum of all possible pairs within each tied group for the first quantity, n_2 is the sum of all possible pairs within each tied group for the second quantity.

τ-c can be written as follows[2]. To obtain m, we first construct the contingency table made from the first and second quantity. Using the rows and columns of the table, m is defined as $\min(\text{num. of rows}, \text{num. of columns})$.

$$\tau_c = \frac{2(n_c - n_d)}{N^2 \frac{m-1}{m}} \tag{4}$$

```
15. deficit:
The company <had a large deficit>.
a: spent a lot more money than it earned
b: went down a lot in value
c: had a plan for its spending
                    that used a lot of money
d: had a lot of money stored in the bank

26. malign:
His <malign> influence is still felt.
a: good
b: evil
c: very important
d: secret
```

Figure 2: Examples of the Vocabulary Size Test. Test-takers are asked to choose the option that paraphrases the part between "<" and ">" from a, b, c, and d.

3 Proposed Method: Vocabulary Testing

This section describes our unsupervised readability assessor that employs a novel approach: instead of using valuable readability labels as the source of text difficulty for typical second language learners, our proposed method uses vocabulary tests as the source of word difficulty for typical second language learners and obtains readability scores based on accurately estimated word difficulty. To this end, this section explains how to analyze vocabulary test result data to obtain word difficulty.

Fig. 2 shows example questions from the vocabulary size test, a widely used vocabulary test in applied linguistics (Beglar and Nation, 2007). Each question asks about a word in a multiple-choice question format. The test consists of 100 questions like those shown in Fig. 2. Ehara (2018) used this test to have 100 second-language learners take the test and to collect their responses. Their data were published and made publicly available. We used their dataset to train our classifiers.

3.1 Evaluating vocabulary test results: Item Response Theory

We want to analyze vocabulary test results to obtain word difficulty values encoding learners' language knowledge. To this end, we employed the idea of *item response theory* (Baker, 2004), a statistical model that can estimate learners' abilities and test

[1] https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient#Tau-b

[2] https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient#Tau-c

questions' difficulties from the learners' responses to the questions.

Let $\mathcal{V}$ be the set of vocabulary, and let $\mathcal{L}$ be the set of learners. Let $z_{v,l} \in \{0, 1\}$ be the result of whether learner $l \in \mathcal{L}$ correctly answered the question for word $v \in \mathcal{V}$: $z_{l,v} = 1$ if l answered correctly for word v; otherwise, $z_{l,v} = 0$. Correct answers usually imply that l knows word v.

Then, by using $\{z_{v,l}\}$ as the training data, we train the following model:

$$p(z = 1|v, l) = \text{sigmoid}(a_l - d_v) \tag{5}$$

In Eq. 5, a_l is the ability parameter of learner l, d_v is the difficulty of word w, and sigmoid denotes the logistic sigmoid function, i.e., $\text{sigmoid}(x) = \frac{1}{1+\exp(-x)}$. The logistic sigmoid function is the binary version of the softmax function, which is frequently used in neural classifiers. It is a monotonously increasing function ranging within $(0, 1)$. As $\text{sigmoid}(0) = \frac{1}{1+1} = \frac{1}{2}$, when a learner's ability a_l is larger than the word difficulty d_v, the probability that learner l knows word v can be written as follows: $p(z = 1|v, l) > \frac{1}{2}$ in Eq. 5. Likewise, by using Eq. 5, we can compare a learner's ability and word difficulty in the same dimension.

To estimate learner ability and word difficulty, $z_{v,l}$ is given as z in Eq. 5 in the training phase. In this way, in *item response theory*, learner ability and word difficulty are comparable, and these parameters are to be estimated from the test result data.

3.2 Obtaining difficulty of words not in the vocabulary test

In Eq. 5, d_v denotes the word difficulty estimated from the vocabulary tests. Here, in addition to the word difficulty for the words within the vocabulary test, we also want to obtain word difficulty values for all words that may appear in the target language. To this end, we calculate d_v from the word frequency in large balanced corpora as follows:

$$d_v = -\sum_{k=1}^{K} w_k \log(\text{freq}_k(v) + 1) \tag{6}$$

In Eq. 6, K is the number of corpora to use, $\text{freq}_k(v)$ denotes the frequency of word v in the k-th corpus, and w_k is the weight parameter of the k-th corpus.

In summary, given the vocabulary test results $\{z_{v,l}\}$ and corpus frequency features $\text{freq}_k(v)$, we can estimate the parameters: namely, the weight of the k-th corpus w_k and learner l's ability a_l. By putting Eq. 5 and Eq. 6 together, we can see that the inside formula of the sigmoid function is linear with respect to the parameters to be estimated because all terms consist of a product of a parameter and a constant calculated from features, and no term has a product of two or more parameters. As the sigmoid function of a linear combination of parameters can be reformulated as a logistic regression, we can implement Eq. 5 and Eq. 6 by using typical logistic regression classifiers such as scikit-learn [3] and LIBLINEAR [4]. We will release our code upon the acceptance of the paper.

Note that we do not use the valuable readability label $\{y_i\}$ in the training phase; hence, our method is categorized as an unsupervised method.

3.3 Proposed Automatic Readability Assessor

After estimating the parameters using the above-mentioned procedure, we use the following formula to obtain the readability of given $\mathcal{T}_i$. Here, l_{avg} denotes the test-taker whose estimated ability parameter is closest to the average of the estimated ability parameter values $\{a_l\}$s. Intuitively, the following equation calculates the probability that the average learner knows all the words that appear in $\mathcal{T}_i$ and uses it as the readability score:

$$\begin{aligned} s_i &= \mathit{score}(\mathcal{T}_i) \\ &= -\frac{1}{|\mathcal{T}_i|} \log \left(\prod_{v \in \mathcal{T}_i} p(z = 1|v, l_{\text{avg}}) \right). \end{aligned} \tag{7}$$

4 Experimental Settings

4.1 Choice of Dataset

We used the OneStopEnglish dataset (Vajjala and Lučić, 2018) for our evaluation because of the following reasons. First, it is one of the newest datasets. Second, it is publicly available and downloadable. Third, it is a reliable dataset in the sense that it has no known pitfalls when used as a corpus for evaluation.

While Martinc et al. (2021) uses other corpora such as the WeeBit corpus (Xia et al., 2016) and the Newsela corpus (Xu et al., 2015), both have known pitfalls when used for the evaluation of automatic readability assessment. The WeeBit corpus is not a *parallel corpus*, which is explained in the next

[3] https://scikit-learn.org/stable/

[4] https://www.csie.ntu.edu.tw/~cjlin/liblinear/

subsection. This means that each level consists of totally different articles covering different topics. As some topics such as politics tend to use more difficult phrases than other topics, it is difficult to see how the topic of content influences the resulting performance values. The Newsela corpus is a parallel corpus, which removes the influence caused by topics. However, according to Martinc et al. (2021), its readability labels can be easily identified from the average sentence length in a text: the average sentence length achieved 0.906 in the Pearson's ρ correlation. Hence, even if a method works well on the Newsela corpus, it could be possible that the method merely inherently calculates and uses average sentence length.

4.2 OneStopEnglish dataset

Regarding the source of the dataset, Vajjala and Lučić (2018) says that "onestopenglish.com is an English language learning resources website run by MacMillan Education, with over 700,000 users across 100 countries."

The dataset has three levels: elementary, intermediate, and advanced. According to Vajjala and Lučić (2018), the original articles were taken from the Guardian newspaper. The OneStopEnglish dataset is a parallel corpus, i.e, language teachers manually rewrote the original articles into the three aforementioned readability levels. Hence, one notable characteristic of this dataset is that all three levels have the same content with different readability levels. Hence, by using this dataset, we can avoid having classifiers learn differences in content or topic rather than readability levels.

All three levels have 189 texts each, 567 texts in total. We split these texts into a *training set* consisting of 339 texts, a *validation* set consisting of 114 texts, and a *test* set consisting of 114 texts. The *training set* and *validation* sets were used to train solely supervised methods for comparison. Unsupervised methods did not use the training and validation sets; they used only the test set.

4.3 Baseline Methods

4.3.1 Supervised methods

First, we introduce the supervised methods that we used for comparison because it involves the training data mentioned right above. As the BERT-based sequence classification has been reported to achieve excellent results (Devlin et al., 2019), we applied the standard BERT-based sequence classification approach involving pretraining and fine-tuning. For the pretrained model, we used **bert-large-cased-whole-word-masking** in the Huggingface models [5].

Then, we fine-tuned the model using the aforementioned 339 training texts. For this fine-tuning, we used a GeForce RTX 3090 board that has 24 GiB of Graphical Processing Unit (GPU) memory. The fine-tuning and resulting model took up 16 GiB of GPU memory. This means that it is difficult to achieve similar performance without GPUs with large memory. We named this fine-tuned model **spvBERT**, in which "spv" denotes being supervised. In order to see how the size of training data has an influence on the performance, we also conducted experiments with 168 training texts, which amounted to almost half of the total 339 training texts. We named this model **spvBERT_half**.

All the fine-tuning procedures were conducted using the Adam optimizer (Kingma and Ba, 2015) with a setting of 10 epochs and a 0.00001 training rate.

4.3.2 Unsupervised methods

For the implementation of conventional readability formulae, we used the **readability** PyPI package [6]. We used almost all readability formulae implemented in this package for our experiments: namely, **Flesch-Kincaid** (Flesch-Kincaid Grade Level, FKGL) (Kincaid et al., 1975), **ARI** (Automated Readability Index) (Senter and Smith, 1967), the **Coleman-Liau** Index (Coleman and Liau, 1975), **Flesch Reading Ease** (Flesch, 1948), the **Gunning Fog Index** (Gunning, 1952), **LIX** (Björnsson, 1968), the **SMOG Index** (Mc Laughlin, 1969), the RIX index (Anderson, 1983), and the **Dale-Chall Index** (Dale and Chall, 1948). Among these methods, notably, some formulae such as the Dale-Chall Index depend on their own list of easy/difficult words. Others, such as the Flesch-Kincaid grade level (FKGL), do not require such a list of difficult words but use superficial features such as the total number of syllables in a text. For space limitation, we do not cite all equations, however, we only cite FKGL as being famous and cite Dale-Chall as showing good performance in our evaluation.

$$\text{FKGL} \quad = \quad 0.39\frac{\text{total words}}{\text{total sentences}} \tag{8}$$

[5] https://huggingface.co/models

[6] https://pypi.org/project/readability/

$$+ \quad 11.8\frac{\text{total syllables}}{\text{total sentences}} - 15.59$$

$$\text{Dale-Chall} \quad = \quad 15.79 \times \left(\frac{\text{difficult words}}{\text{words}}\right) \quad (9)$$
$$+ \quad 0.0496\left(\frac{\text{words}}{\text{sentences}}\right)$$

More details of these formulae and their implementation are described on the project page. All of these readability formulae are *unsupervised* in the sense that they do not require any training data.

For the unsupervised neural language model, we also used the **bert-large-cased-whole-word-masking** pretraining model and used the **BertForMaskedLM** function to obtain the perplexity of each sentence of the text of interest. We chose this pretraining model because Martinc et al. (2021) reported that they used **bert-base-uncased** and reported not so good performance, so we chose a BERT-based model larger than the one that they used. Note that, unlike neural sequence classification, language models are designed to be unsupervised and thus do not require any training data to fine-tune. All we need to do for the neural language model is to input each sentence in the text of interest and calculate the perplexity score of the inputted sentence.

For splitting a text into sentences, we used the **sent_tokenize** function in the **nltk** Python package [7]. After the split, we simply used the average of the perplexity scores of each sentence in a text as the readability score. As the perplexity score of a sentence encodes the fluency of the inputted sentence, this roughly measures the overall fluency of the inputted sentence. We call this method **BERTLMavg**, where LM denotes a *language model*.

As **BERTLMavg** does not use fine-tuning, it uses less GPU memory compared to **spvBERT**. However, **BERTLMavg** uses 1,793 MiB of GPU memory to output perplexity scores, which is still impractical in a low-computational-resource environment.

According to Martinc et al. (2021), BERT language models do not perform good results. Hence, while not directly comparable because we could not obtain their test set, for a rough comparison, we cited their best model on the OneStopEnglish dataset, **TCN RSRS-simple**. The model is temporal convolutional network (TCN) trained on the Simplified Wikipedia corpus. For space limitations, refer to Martinc et al. (2021) for the details of this method.

Proposed model was trained on a previously published and publicly available vocabulary dataset (Ehara, 2018). For the corpus word frequency, we used the frequencies taken from the British National Corpus (BNC Consortium, 2007) and the Corpus of Contemporary American English (COCA) (Davies, 2008). Both corpora are balanced general corpora used extensively in English education (Nation, 2006). Especially, the word frequencies of these corpora are important resources for determining word difficulty in English education. For counting text frequencies, we used **nltk.stem.WordNetLemmatizer** in the **nltk** package to lemmatize words appearing in running texts.

Our **Proposed** model uses the average of the negative log likelihood that an average learner knows each word in the text as presented in Eq. 7. As our **Proposed** model uses the BNC and COCA word frequencies, it could be possible that these word frequencies have an essential influence on the performance of the **Proposed** model. To check this, we also measured the correlation between the gold labels and the average negative log of the unigram probability values of the given text in each corpus. We name these feature-based methods as **BNC** and **COCA**.

4.4 Experimental Results: Pearson's ρ and performance

This subsection describes the experimental results showing the problem of using Pearson's ρ in evaluation.

Tab. 1 shows the experimental results. The columns of Tab. 1 show the rank correlation coefficients introduced in the previous sections. Namely, they are Spearman's ρ, Kendall's τ with tie correction type b (τ-b), and Kendall's τ with tie correction type c (τ-c). Pearson's ρ is shown in the rightmost column. As we explained in previous sections, Pearson's ρ is affected by the linearity of scores. To see how Pearson's ρ is affected by the linearity of scores, below each unsupervised method **M**, we show $exp(\mathbf{M})$ to indicate the resulting performance values when we replaced the scores of M with the exponentilized the scores of M, i.e, $\exp(\text{the score of M})$ to remove linearity. The distinction of "unsupervised" and "supervised" is clearly marked in the leftmost column.

In Tab. 1, we can easily see that, for all unsuper-

[7] nltk.org

Super-vision	Method	Spearman's ρ	Kendall's τ-b	Kendall's τ-c	Pearson's ρ
Unsuper-vised	**Flesch-Kincaid**	0.324	0.253	0.308	0.359
	exp(**Flesch-Kincaid**)	0.324	0.253	0.308	0.149
	ARI	0.317	0.248	0.302	0.351
	exp(**ARI**)	0.317	0.248	0.302	0.136
	Coleman-Liau	0.373	0.295	0.359	0.372
	exp(**Coleman-Liau**)	0.373	0.295	0.359	0.185
	FleschReadingEase	-0.387	-0.301	-0.366	-0.426
	exp(**FleschReadingEase**)	-0.387	-0.301	-0.366	-0.169
	GunningFogIndex	0.331	0.257	0.313	0.362
	exp(**GunningFogIndex**)	0.331	0.257	0.313	0.151
	LIX	0.348	0.273	0.332	0.383
	exp(**LIX**)	0.348	0.273	0.332	0.129
	SMOGIndex	0.456	0.360	0.438	0.479
	exp(**SMOGIndex**)	0.456	0.360	0.438	0.306
	RIX	0.437	0.340	0.414	0.462
	exp(**RIX**)	0.437	0.340	0.414	0.181
	DaleChallIndex	0.495	0.387	0.472	0.506
	exp(**DaleChallIndex**)	0.495	0.387	0.472	0.431
	TCN RSRS-simple	-	-	-	0.615(*)
	BERTLMavg	-0.220	-0.173	-0.210	-0.040
	exp(**BERTLMavg**)	-0.220	-0.173	-0.210	-0.005
	BNC	-0.012	-0.009	-0.010	-0.006
	exp(**BNC**)	-0.012	-0.009	-0.010	-0.123
	COCA	-0.018	-0.016	-0.020	-0.039
	exp(**COCA**)	-0.018	-0.016	-0.020	-0.121
	Proposed	**0.730**	**0.592**	**0.709**	**0.715**
	exp(**Proposed**)	**0.730**	**0.592**	**0.709**	0.260
Super-vised	**spvBERT_half**	0.751	0.729	0.725	0.747
	spvBERT	**0.866**	**0.856**	**0.854**	**0.864**

Table 1: Experimental Results on the OneStopEnglish Dataset. For a method M, exp(M) denotes the correlations between the array of exp(M's score) and the gold labels. (*) denotes that the value is cited from other papers.

vised methods except for **BNC** and **COCA**, the correlations of the exponentialized scores measured by Pearson's ρ are closer to 0 than their original scores. In contrast, the rank correlation coefficient values are kept unchanged because exp is a monotonous function and hence the ranking is not altered by the use of exp. The reason why the performance values of **BNC** and **COCA** seem slightly increased is presumably because of noise: **BNC** and **COCA** did not correlate with the readability labels statistically significantly in the first place. Neither did exp(**BNC**) and exp(**COCA**).

The drop in performance scores is enormous for some methods such as **Proposed**: its performance was originally 0.715 but plunges to 0.260 by using exp. This result indicates the vulnerability of using Pearson's ρ in the evaluation: the evaluation by Pearson's ρ is strongly affected by how linear the scores are, suggesting the use of rank correlation for better evaluation.

TCN RSRS-simple is the best model using the same dataset in Martinc et al. (2021). As they show only the performance measured by the Pearson correlation, we wrote – for other rank correlation coefficients. Also note that we cannot make direct comparison as we could not obtain their test set used for their experiments. This is marked by the (*) after the value. While we can see that **Proposed** achieved better correlation than **TCN RSRS-simple**, we are not sure if this result indicates the linearity of the methods or the superiority of **Proposed** against **TCN RSRS-simple**. Likewise, the use of Pearson's ρ only makes followup papers' efforts to compare results difficult.

4.5 Performance Comparison

In all unsupervised methods, our **Proposed** method achieved the best results in all rank correlation coefficients and Pearson's ρ, although we need to be careful with the interpretation of Pearson's rho as explained in Sec. 2.2. These results were statistically significant ($p < 0.01$): all correlation coefficients can also be used for statistical testing. In each of the statistical tests, the null hypothesis is that no association exists between the scores and the gold labels. When measured using Spearman's ρ, **Proposed** achieved a value of 0.730, which is close to 0.751, the performance achieved by *supervised* BERT using half of the training data.

BERTLMavg did not achieve good results in predicting readability labels. This result suggests that perplexity and readability are different measures and that, to measure readability, we need to obtain and make use of the information about what a typical language learner knows about the target second language.

Interestingly, **BNC** and **COCA** achieved poor results in predicting readability labels. This result shows that the reason that **Proposed** method outperformed the others is not merely because the features that **Proposed** used are excellent. A good combination of the two features results in significant results. The use of only one of the two does not achieve good results. Hence, we can see that **Proposed** works excellently for making the combination of the two corpus-based features.

For a comparison with supervised models, Tab. 1 shows their performances: **spvBERT** and **spvBERT_half**. Supervised models output labels rather than scores in their prediction phase: we directly used these labels to calculate rank correlation coefficients for a fair comparison with unsupervised models. Leveraging the supervision, they outperformed most of the unsupervised methods in all rank correlation coefficients. This means that using valuable supervision yields great improvement in the predictive performance of readability.

The performance differences among **spvBERT**, **BERTLMavg**, and **Proposed** can be interpreted as follows. BERT is a large model trying to use as much information as possible from a sentence, such as syntactic structure. Hence, it is difficult for the model to find useful information contributing to readability without supervision. **Proposed** is a bag-of-words model that is designed to be lightweight by sacrificing such complicated factors. Hence, the performance difference between **spvBERT** and **Proposed** can be regarded as a degree that information beyond word difficulty – such as syntactic information or sentence context – accounts for readability. While this is beyond the focus of this paper, a detailed error analysis between **spvBERT** and **Proposed** may lead to understanding what kind of syntactic information or contexts in a sentence contribute to readability.

4.6 Memory and Speed

We used a Core i7-10700K (3.80 GHz) machine with a GeForce RTX 3090 board for all experiments. The **BERTLMavg**, which is an unsupervised BERT language model, uses $1,793$ MiB GPU memory. In contrast, **Proposed** is merely a logistic regression and does not require as GPU for practical use. In addition, the model's features are smaller than those of the BERT models. **Proposed** uses the BNC and COCA frequencies, which amount to 10 MiB of CPU memory, which is roughly $\frac{1}{100}$ of that used by the unsupervised BERT models. In terms of speed, to classify all texts in the test set, while **BERTLMavg** utilizes 368 s, **Proposed** utilizes only 5.37 s. This indicates that the **Proposed** is 68.5 times faster than **BERTLMavg**.

5 Discussion

In this paper, we discussed the necessity to use rank correlation coefficients to evaluate automatic readability assessment. The problem that Pearson correlation coefficients reflect not only the correlation between two scores but also the linearity of the scores is not particularly novel and has been pointed out for a long time. This study showed that this problem has a significant impact in the evaluation of the ARA task. In fact, in the recent evaluation of the ARA task (Martinc et al., 2021), the problem of linearity in the Pearson coefficients was *not* addressed and its evaluation simply uses the Pearson correlation coefficients. To the best of our knowledge, this is the first study to demonstrate the effect of this problem on the performance values in the ARA task and examine the extent to which the linearity of the scores affects the scores in Tab. 1.

In this paper, we termed the **Proposed** method as "unsupervised," according to (Martinc et al., 2021). They termed methods that do not use manually annotated readability labels as "unsupervised" even if the methods use supervised machine learning. In fact, the proposed method is trained using the vocabulary test dataset (Ehara, 2018). Phrases, such as "the most average learner" and "learner ability, all refer to the learners on this vocabulary test dataset. In this study, knowing that the term supervised/unsupervised is misleading, we deliberately described the proposed method as "unsupervised" for easy comparison with previous studies.

In NLP, the **Proposed** method is closely related to complex word identification (CWI) tasks (Yimam et al., 2018; Paetzold and Specia, 2016). CWI is a task that aims to discover difficult words in a text. The relationship between CWI and personalized text readability was previously studied in (Ehara, 2019). The task of obtaining the difficulty of an English word for each individual ESL learner, as we did in this study, can be regarded as personalized CWI (Ehara et al., 2012, 2014) [8]. Personalized CWI has many downstream applications in NLP such as lexical simplification (Lee and Yeung, 2018, 2019), text recommendation for language learners (Ehara et al., 2013; Yeung and Lee, 2018; Lee, 2021), and translator selection in crowdsourcing (Ehara et al., 2016). Some studies focus on the relationship between word semantics and word difficulty (Ehara et al., 2014; Beinborn et al., 2016; Ehara, 2020b). Regarding the interpretability of CWI classifiers, Ehara (2020a) studied the relationship CWI classifiers' weights and vocabulary sizes.

6 Conclusions

In this paper, we investigated the correlation coefficients to evaluate the performance of unsupervised automatic readability assessors. The experimental results showed that the readability performances measured by Pearson's ρ are strongly affected by the linearity of the output scores, whereas those measured by rank correlations are not affected. This indicates the appropriateness of using rank correlation coefficients to evaluate unsupervised automatic readability assessors. We also proposed a lightweight unsupervised assessor based on word difficulty for typical second language learners calculated from a vocabulary test result dataset. This assessor could achieve the best score among all the compared unsupervised assessors.

In the future, we plan to conduct a more detailed analysis to investigate which rank correlations, including those not introduced in this paper, are more appropriate for the evaluation.

Acknowledgment

This study was supported by JST ACT-X Grant Number JPMJAX2006 and JSPS KAKENHI Grant Number 18K18118. We used the ABCI infrastructure from AIST for the computational resources. We appreciate anonymous reviewers for their valuable comments.

References

Mayer Alvo and LH Philip. 2014. *Statistical methods for ranking data*, volume 1341. Springer.

Ilaria L Amerise and Agostino Tarsitano. 2015. Correction methods for ties in rank correlations. *Journal of Applied Statistics*, 42(12):2584–2596.

Jonathan Anderson. 1983. Lix and rix: Variations on a little-known readability index. *Journal of Reading*, 26(6):490–496.

Frank B. Baker. 2004. *Item Response Theory : Parameter Estimation Techniques, Second Edition*. CRC Press.

David Beglar and Paul Nation. 2007. A vocabulary size test. *The Language Teacher*, 31(7):9–13.

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2016. Predicting the Spelling Difficulty of Words for Language Learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 73–83, San Diego, CA. Association for Computational Linguistics.

C. H. Björnsson. 1968. Läsbarhet, Stockholm.

BNC Consortium. 2007. The british national corpus, version 3 (bnc xml edition). Distributed by Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium http://www.natcorp.ox.ac.uk/.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Mark Davies. 2008. The corpus of contemporary american english (coca). Available online at https://www.english-corpora.org/coca/.

[8] The journal version of (Ehara et al., 2012) is (Ehara et al., 2018).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL*, pages 4171–4186, Minneapolis, Minnesota.

Yo Ehara. 2018. Building an English Vocabulary Knowledge Dataset of Japanese English-as-a-Second-Language Learners Using Crowdsourcing. In *Proc. of LREC*.

Yo Ehara. 2019. Uncertainty-Aware Personalized Readability Assessments for Second Language Learners. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1909–1916.

Yo Ehara. 2020a. Interpreting neural CWI classifiers' weights as vocabulary size. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 171–176, Seattle, WA, USA → Online. Association for Computational Linguistics.

Yo Ehara. 2020b. Neural rasch model: How do word embeddings adjust word difficulty? In *Computational Linguistics – 16th International Conference of the Pacific Association for Computational Linguistics, PACLING 2019, Hanoi, Vietnam, October 11–13, 2019, Revised Selected Papers*, pages 88–96, Singapore. Springer Singapore.

Yo Ehara, Yukino Baba, Masao Utiyama, and Eiichiro Sumita. 2016. Assessing Translation Ability through Vocabulary Ability Assessment. In *Proc. of IJCAI*.

Yo Ehara, Yusuke Miyao, Hidekazu Oiwa, Issei Sato, and Hiroshi Nakagawa. 2014. Formalizing Word Sampling for Vocabulary Prediction as Graph-based Active Learning. In *Proc. of EMNLP*, pages 1374–1384.

Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2012. Mining Words in the Minds of Second Language Learners: Learner-Specific Word Difficulty. In *Proceedings of COLING 2012*, pages 799–814, Mumbai, India. The COLING 2012 Organizing Committee.

Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2018. Mining Words in the Minds of Second Language Learners for Learner-specific Word Difficulty. *Journal of Information Processing*, 26:267–275.

Yo Ehara, Nobuyuki Shimizu, Takashi Ninomiya, and Hiroshi Nakagawa. 2013. Personalized Reading Support for Second-language Web Documents. *ACM Trans. Intell. Syst. Technol.*, 4(2):31:1–31:19.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A Comparison of Features for Automatic Readability Assessment. pages 276–284.

Rudolf Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

Robert Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

John Lee and Chak Yan Yeung. 2018. Personalizing Lexical Simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 224–232, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

John Lee and Chak Yan Yeung. 2019. Personalized Substitution Ranking for Lexical Simplification. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 258–267, Tokyo, Japan. Association for Computational Linguistics.

John SY Lee. 2021. An editable learner model for text recommendation for language learning. *ReCALL*, pages 1–15.

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 Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Mavuto M Mukaka. 2012. A guide to appropriate use of correlation coefficient in medical research. *Malawi medical journal*, 24(3):69–71.

I. Nation. 2006. How Large a Vocabulary is Needed For Reading and Listening? *Canadian Modern Language Review*, 63(1):59–82.

Gustavo Paetzold and Lucia Specia. 2016. Collecting and Exploring Everyday Language for Predicting Psycholinguistic Properties of Words. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1669–1679, Osaka, Japan. The COLING 2016 Organizing Committee.

RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, CINCINNATI UNIV OH.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In

Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text Readability Assessment for Second Language Learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, San Diego, CA. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Chak Yan Yeung and John Lee. 2018. Personalized Text Retrieval for Learners of Chinese as a Foreign Language. In *Proc. of COLING*, pages 3448–3455.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo H. Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A Report on the Complex Word Identification Shared Task 2018. *arXiv:1804.09132 [cs]*. ArXiv: 1804.09132.