Electronic Preprint for Journal of Information Processing Vol.30

Regular Paper

Uncertainty-aware Personalized Readability Assessment Framework for Second Language Learners

Yo Ehara[1,a)]

Received: August 24, 2021, Accepted: February 4, 2022

Abstract: Assessing whether an ungraded second language learner can read a given text quickly is important for supporting learners of diverse backgrounds. Second language acquisition (SLA) studies have tackled such assessment tasks wherein only a single short vocabulary test result is available to assess a learner. Such studies have shown that the text-coverage or namely the percentage of words the learner knows in the text, is the key assessment measure. Currently, count-based percentages are used, in which each word in the given text is classified as being known/unknown to the learner, and the words classified as known are then simply counted. When each word is classified, we can also obtain an uncertainty value as to how likely each word is known to the learner. However, how to leverage these informative values to guarantee their use as an assessment measure that is comparable to that of the previous values remains unclear. We propose a novel framework that allows assessment methods to be uncertainty-aware while guaranteeing comparability to the text-coverage threshold. Such methods involve a computationally complex problem for which we also propose a practical algorithm. In our evaluation using newly created crowdsourcing-based dataset, our best method under our framework outperformed conventional methods.

Keywords: uncertainty, vocabulary tests, readability assessments, natural language processing

1. Introduction

Second language learners, particularly adults, have diverse backgrounds. They may have different first languages, they may have started learning from different ages, and they may have undergone different styles of education. Despite their diverse backgrounds, there are many social situations in which we need to quickly assess whether each learner can read a text; for example, when choosing the first textbook for each newcomer to a language school or screening immigrants who may need language assistance for reading administrative documents that they need to understand.

In such situations, a learner assessment should be finished quickly using the minimum manual effort required for an evaluation. Previous studies [15], [23], [26] have shown that vocabulary tests meet such conditions. Typically, such a test consists of multiple-choice questions that can be scored easily, and a learner can finish answering 100 questions in only about a half an hour. Given one quick vocabulary test result for a learner and a text of interest, our goal is to assess whether the learner can read the text. Owing to their diverse backgrounds, it may not be possible to classify learners with a one-dimensional ability scale, i.e., some learners can be good at some particular types of words, for example, musicians can be expected to know more music-related words in the test than others. Hence, our goal is not to measure learners' ability, but to assess whether each learner can read a given text. We call this task Personalized Readability Assessment (PRA) (**Fig. 1**).

[1] Tokyo Gakugei University, Koganei, Tokyo 184–8501, Japan.
[a)] ehara@u-gakugei.ac.jp

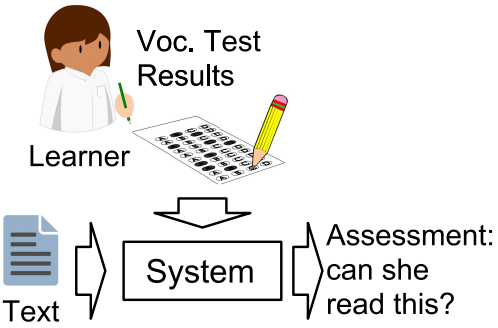


Fig. 1 Task setting for PRA. Given a learner's vocabulary test result, the task is to assess whether the learner can read a text.

Previous studies on a PRA have independently shown that the *text coverage* (TC), namely, the percentage of words that the learner knows in a text, is a key measure to assess readability [15], [23]. To calculate this percentage using the vocabulary test result of each learner, these methods first classify each word in the text based on whether it is known to the learner. Because this is merely a binary classification, there are numerous methods that are applicable. If the count-based percentage of words classified as known, is above a certain threshold, the text is assessed to be readable to the learner. Interestingly, this threshold is known to have a relatively small range of 95%–98% for various texts and diverse learners.

In education, it is common to improve the ability of learners by having them read texts that are slightly more difficult than their ability. In order to do this, it is useful to be able to measure the probability that the learner will be able to read the text at a level close to the learner's ability. This enables us to build a system that can search texts that the learner may not be able to read and

fully understand and hence is appropriate for their learning the language.

Because this assessment is based on classifying each word as known or unknown to a learner, improving this classification is of key importance. When classifying each word in a text, we can obtain not only a binary classification but also a classification of the *uncertainty values*, demonstrating the confidence of classification which is typically within the range of [0,1]. Because we have only one quick vocabulary test result available for each learner, each classification can be uncertain for numerous words, e.g., 0.4 or 0.6. Leveraging these values is reasonably expected to improve the assessment accuracy and may provide more detailed information for each learner. These uncertainty values, however, have previously been simply ignored and converted into either a 0 or 1 because it was unclear how to leverage these values to calculate an assessment measure comparable to the text-coverage. Without a comparable measure, we cannot utilize previously determined text-coverage threshold values for an assessment.

To this end, we propose a novel uncertainty-aware framework for PRA. In our framework, assessment methods can leverage uncertainty values from classifiers because the approach generalizes text-coverage to uncertainty-aware measures comparable to the previously validated thresholds. Calculating generalized text-coverage values requires solving a computationally complex problem for which we also propose a practical algorithm.

Evaluation of a real dataset created using crowdsourcing showed that the best method derived from our framework which leverages uncertainty values, outperformed conventional methods by a maximum of 11 points in terms of the mean average precision when assessing personalized readability. Moreover, qualitative results showed that methods under our framework can indicate whether we need more information to accurately assess a learner.

Our contributions are as follows:

(1) To assess whether a learner can read a text based only on a single vocabulary test result, we propose a novel framework that allows methods to be uncertainty-aware while keeping their assessment results comparable to the count-based text-coverage thresholds used in previous studies.
(2) The derived methods involve computationally complex problems for which we also propose a practical dynamic-programming-based algorithm.
(3) To evaluate the accuracy of the derived methods, we created an openly available personalized readability dataset using a crowdsourcing method [*1].
(4) The evaluation results show that the best method derived here consistently outperforms the conventional methods in terms of the mean average precision by a maximum of 11 points. The derived methods can also provide a detailed qualitative analysis.

2. Related Work

Readability assessment has been studied in natural language processing (NLP) as well as second language acquisition (SLA). However, such studies typically assume that more information is available to the learner such as syntactic or reading comprehension test results [2], [18], [32]. While such information may lead to accurate assessments, it typically requires skilled evaluators and more time to assess and so does not meet our need to quickly assess readability. Hence, our study is focused on vocabulary or lexical aspects.

In addition, most NLP readability assessment tasks are not personalized since these assume that learners are already graded. In such cases, predicting the grade for a given text is the main focus of the readability assessment. In second-language vocabulary tutoring studies, while personalizing affective policy has been addressed [14], this study does not address readability assessments. A few studies focusing on personalized readability assessment in NLP have been published, including complex word identification (CWI) task studies [11], [24]. CWI is a task that aims to identify complex words in a text [28], [34]. Unlike personalized CWI, which identifies only complex words, our goal is to assess whether the target learner can read a given text using personalized CWI classifiers.

Personalized readability assessment has a direct application in retrieving appropriate learning materials for learners. PRA was applied for this purpose in English [7], [9], [10], [11], [12] and later in Chinese [33] using conventional uncertainty-unaware methods.

With the aim of quickly assessing learners using vocabulary tests, the PRA task became the main focus in English second language acquisition studies [15], [22]. To calculate text-coverage, some of these early studies conducted reading comprehension tests in which test-takers were also asked to report the words that they did not know in the texts of the tests. Soon, it was reported that a text is readable if a learner knows 95% or more of the words in the text.

Studies that followed this earlier research proposed more convenient methods for classification or reported more detailed thresholds. The study [27] proposed a carefully designed vocabulary test called vocabulary size test (VST) based on the frequency ranking of the British National Corpus [3]. This test consists of 100 questions each of which consists of a sentence containing the target word and asks the test-taker to choose the correct meaning of the word from multiple options. The questions are randomly sampled from the most frequent 20,000 words while carefully avoiding misleading or confusing words. Then, by simply multiplying 200 by the number of correctly answered questions, the vocabulary size of each learner can be estimated.

By classifying each word in a text, their method is based on a rough assumption that, in the ranking order of the 20,000 words, based on the learner's vocabulary size, the learner knows all of the more frequently used words and does not know all of the less frequently used words. Clearly, this is not a realistic assumption, and it was reported that this only works when the backgrounds of the learners are not very diverse, such as with students attending the same school [27]. We call this method **VST**.

To achieve a more accurate classification, previous NLP studies have proposed machine-learning-based models [11], [24]. Given a learner and a word, in these studies, they train a binary

[*1] Refer to Section A.2 for our dataset and code.

classifier that models the ability of each learner as a parameter and uses it to classify each word in the text. Although uncertainty values are obtainable from some of their models, the authors did not address how to appropriately handle these uncertainties.

Notably, PRA methods can be easily exported to languages other than English because assessments are solely based on vocabulary test results without using other complex linguistic features such as syntax. PRA methods have been reported to be successful in both Chinese and Japanese [4], [25].

In NLP, outside of the PRA task, a mathematically similar problem has been studied for the very different aim of aggregating document classifier outputs to estimate the relative frequency over document labels. These are known as *prevalence estimation* tasks [19].

3. Formalization

This section introduces our novel uncertainty-aware framework. We first introduce the conventional PRA methods.

First, we define the notation. Our goal is to assess whether learner l_j can read text $\mathcal{T}$. We consider J learners, i.e., $\{l_1, \ldots, l_J\}$. We let text $\mathcal{T}$ consist of vocabulary with size I, i.e., $\{v_1, \ldots, v_I\}$. We denote f_i as the frequency of word v_i in text $\mathcal{T}$. Therefore, the total frequency of the words in $\mathcal{T}$ is $|\mathcal{T}| = \sum_{i=1}^{I} f_i$. Let $y_{i,j} \in \{0, 1\}$ be a binary variable that takes 1 when learner l_j knows word v_i; otherwise, apply a value of 0.

The notation used here is summarized in **Table 1**. Using this notation, the text-coverage of learner l_j in text $\mathcal{T}$, $C_{\mathcal{T},j}$, is calculated as follows.

$$C_{\mathcal{T},j} = \frac{\sum_{i=1}^{I} y_{i,j} f_i}{|\mathcal{T}|} = \sum_{i=1}^{I} y_{i,j} \frac{f_i}{|\mathcal{T}|} \tag{1}$$

As stated above, it must be determined whether the TC defined in Eq. (1) is larger than the TC *threshold* δ. That is, it is necessary that $C_{\mathcal{T},j} \geq \delta$ for learner j to be able to read text $\mathcal{T}$. Here, δ usually ranges from 0.95 to 0.98 [23], [26].

In previous studies, $y_{i,j}$ in Eq. (1) has been considered as merely a binary variable and not a *random* binary variable. This means that we assume that the TC is *deterministic* without *uncertainty*. This assumption is unrealistic for the following reason. It is difficult to practically test learner j based on all the words possibly occurring in a text. In practice, even for a learner whose second language vocabulary is scrutinized, some words in $\mathcal{T}$ are likely to remain untested and therefore uncertain. Thus, by regarding $y_{i,j}$ as a random variable rather than a fixed constant, more realistic situations can be better modeled in second-language learning.

Table 1 Definitions of variables.

j	learner index
l_j	j-th learner. J is the number of learners.
$\mathcal{T}$	text to read.
i	index in the vocabulary of $\mathcal{T}$. The size of the vocabulary is I.
v_i	i-th word in the vocabulary of $\mathcal{T}$.
f_i	frequency of i-th word v_i in $\mathcal{T}$.
$\|\mathcal{T}\|$	total number of words in $\mathcal{T}$. $\|\mathcal{T}\| = \sum_{i=1}^{I} f_i$.
δ	Text-coverage threshold. $0 \leq \delta \leq 1$.
$C_{\mathcal{T},j}$	Text-coverage of learner j in text $\mathcal{T}$.
$y_{i,j}$	$\in \{0, 1\}$. 1 if word v_i is classified as known to learner l_j; otherwise, 0.
$p_{i,j}$	probability, i.e., uncertainty, that $y_{i,j} = 1$. $P(y_{i,j} = 1)$. We also write p_i when j is fixed.
S_j	$\sum_{i=1}^{I} y_{i,j} f_i$.

3.1 Proposed Uncertainty-aware Framework

The key to our uncertainty-aware framework is to regard $y_{i,j}$ in Eq. (1) as a binary *random* variable. Specifically, we model $y_{i,j}$ using a Bernoulli distribution $y_{i,j} \sim B(p_{i,j})$, where $P(y_{i,j} = 1) = p_{i,j}$.

Let us revisit the text-coverage defined in Eq. (1). Note that this text-coverage is defined as merely a weighted sum of $y_{i,j}$ weighted using the normalization constant $\frac{f_i}{|\mathcal{T}|}$. Because the sum of random variables with constant weights is another random variable, if we regard $y_{i,j}$ as a random variable, the *text-coverage is also a random variable*. Hence, Eq. (1) can be viewed as follows: We have I binary random variables $\{y_{1,j}, \ldots, y_{I,j}$, independently but *not identically* distributed, and we need to calculate the probability that the weighted sum of these random variables is above the threshold $|\mathcal{T}|\delta$. By writing the weighted sum as $S_j = \sum_{i=1}^{I} y_{i,j} f_i$ for simplicity, this can be written as follows.

$$P(C_{\mathcal{T},j} \geq \delta) = P\left(S_j \geq |\mathcal{T}|\delta\right) \tag{2}$$

3.2 Generality

Equation (2) showed that the event in which the text coverage surpasses threshold δ is now a probabilistic event. This occurs with the probability of $P(C_{\mathcal{T},j} \geq \delta)$, and may not occur under $1 - P(C_{\mathcal{T},j} \geq \delta)$.

Our framework Eq. (2) is a generalization of the previous framework. The methods applied in the previous framework can be regarded as a special case of Eq. (2) when $P(C_{\mathcal{T},j} \geq \delta)$ and each $P(y_{i,j} = 1) = p_{i,j}$ is exactly 0 or 1.

As summarized in Section 2, the methods used with the previous framework classify each word i as known or unknown to learner j or in other words determine the value of $y_{i,j}$ as 0 or 1. In our framework, this can be equivalently interpreted as determining the value of $P(y_{i,j})$ as follows: Determining $y_{i,j}$ as 1 is equivalent to determining $P(y_{i,j} = 1) = 1$, and determining $y_{i,j} = 0$ is equivalent to determining $P(y_{i,j} = 1) = 0$.

3.3 Complexity of Eq. (2)

In Eq. (2), let us focus on $\sum_{i=1}^{I} y_{i,j} f_i$. Each $y_{i,j}$ follows a Bernoulli distribution and is weighted by the constant f_i. Hence, its distributioni is called a weighted *Poisson-binomial distribution*, or "the distribution of a sum of independent Bernoulli random variables which may have non-equal expectations" [5]. In other words, it is a sum of independent but not identically distributed Bernoulli random variables. Notably, whereas it is known that a sum of independent-and-*identically*-distributed Bernoulli random variables follows a binomial distribution, this does not follow a binomial distribution because each $y_{i,j}$ is not identically distributed.

How can we calculate the probability that S_j is above the threshold $|\mathcal{T}|\delta$ in Eq. (2)? Naively, we can calculate it by enumerating all possible values of $\{y_{1,j}, y_{2,j}, \ldots, y_{I,j}\}$ starting from $\{0, 0, \ldots, 0\}$ to $\{1, 1, \ldots, 1\}$ and sum the probability of each of

these cases by checking if $\sum_{i=1}^{I} y_{i,j} f_i$ is above the threshold. However, we immediately notice that this method is computationally prohibitive because of the exponential cost of enumerating the combinations $O(2^I)$.

4. Proposed Algorithm Using Subset-sum

Equation (2) formalizes the probability denoting how likely it is that the text-coverage surpasses δ (delta). However, calculating this probability is not straightforward since the naive method is computationally prohibitive.

To this end, we propose an exact and practical algorithm to obtain the probability of Eq. (2). Our key focus in Eq. (2) is that f_i, the frequency of word v_i in text $\mathcal{T}$, is an *integer*. Hence, we can see that S_j only takes an integer value when $y_{i,j}$ takes a value of 0 or 1.

Unlike continuous values, we can iterate through integers. Let us recall $\delta \in [0, 1]$ as a rate. Let us simply write the smallest integer that is equal to or larger than the threshold $|\mathcal{T}|\delta$ as $M = \lceil |\mathcal{T}|\delta \rceil$. Then, Eq. (2) can be simply written as the sum of probabilities from $S_j = M$ to $S_j = |\mathcal{T}|$.

$$P(S_j \geq |\mathcal{T}|\delta) = P(S_j = M) + P(S_j = M + 1) + \cdots \\ \cdots + P(S_j = |\mathcal{T}|) \quad (3)$$

Equation (3) shows that, to calculate the probability of Eq. (2), we only need to calculate $P(S_j = N)$ for a given integer N. Interestingly, the calculation of $P(S_j = N)$ can be reduced to a modified version of the famous *subset-sum* [20] problem as follows.

Let us revisit the definition of S_j, i.e., $S_j = \sum_{i=1}^{I} y_{i,j} f_i$. Here, S_j is the sum of f_is whose $y_{i,j} = 1$. Hence, we can regard S_j as a sum of f_is, where $y_{i,j}$ determines whether f_i is included in the sum. We can restate this using the following *subsets*: Let us denote $\mathcal{F} = \{f_1, f_2, \ldots, f_I\}$ for simplicity. In addition, consider a subset of $\mathcal{F}$, namely, $\mathcal{A} \subseteq \mathcal{F}$. Then, determining the values of $y_{1,j}, y_{2,j}, \ldots$, and $y_{I,j}$ is identical to choosing a subset $\mathcal{A}$ such that $y_{i,j} = 1$ if and only if $f_i \in \mathcal{A}$, and $y_{i,j} = 0$ if and only if $f_i \notin \mathcal{A}$. For example, if $y_{1,j}$ and $y_{3,j}$ are 1 and all the other values of $y_{i,j}$ are 0, then $\mathcal{A} = \{f_1, f_3\}$. Hence, our goal is to first find *subsets* of $\mathcal{F}$ that exactly sum to a given positive integer m, and to then calculate the probability of each of such subset arising.

The first problem, finding subsets of $\mathcal{F}$ that exactly sum to a given integer m, is exactly what the famous subset-sum problem deals with. Although this is an *NP-complete* problem, a popular practical dynamic programming (DP) based algorithm can be applied [20]. Unlike the first problem, the second problem, calculating the probability that each such subset arises, seems to have been not so popularly addressed in Ref. [20]. We found that this probability can also be calculated by modifying the DP-based algorithm for the subset-sum problem, as implemented with *SubsetSumP* in Algorithm 1.

Algorithm 1 operates as follows. First, the algorithm takes the listed inputs. Our goal is to obtain the probability that the text-coverage of $\mathcal{T}$ surpasses δ, given the index of learners of interest j, text $\mathcal{T}$, its word frequencies $\{f_1, f_2, \ldots, f_I\}$, and the probability that learner l_j knows each word $\{p_1, p_2, \ldots, p_I\}$. Algorithm 1 is composed of two functions. The first, *ProbTCsurpass*, is the main function returning $P(S_j \geq |\mathcal{T}|\delta)$. Internally, according to Eq. (3), it repeatedly calls the second, *SubsetSumP*, which eventually calculates $P(S_j = N)$ for a given integer N.

SubsetSumP takes two augments: i and N. It returns the probability that the subsets of $\{f_1, f_2, \ldots, f_i\}$ will exactly sum to N. Notably, when $i = I$, *SubsetSumP*(I, N) returns $P(S_j = N)$. Internally, *SubsetSumP*(I, N) performs a calculation by recursively calling itself. Whereas DP algorithms are typically described using so-called DP tables to remove the need for recursive calls, we did not use such tables to describe Algorithm 1 because recursive calls are easier to understand. Although recursive calls can be slow if the overhead incurred when calling a function is large, in practice, by using the *memoization technique*, or simply caching the arguments and returned values of *SubsetSumP* in a hash table, *SubsetSumP* operates at a practical speed. In *SubsetSumP*, the probability that $\{f_1, f_2, \ldots, f_i\}$ sums up to N is recursively expressed as the sum of the probabilities of summing up $\{f_1, f_2, \ldots, f_{i-1}\}$ in Line 17 and Line 18.

4.1 Computational Complexity

As described, our algorithm leverages Eq. (3) and solves the subset-sum problem multiple times to obtain $P(S_j = N)$. Because the subset-sum problem is known as an NP-complete problem [20], this approach is also at least NP-complete.

The DP-based algorithm, which solves the subset-sum problem in a practical manner, has a *pseudo-polynomial time* complexity [20]. Since this algorithm and our *SubsetSumP* function in Algorithm 1 essentially differ only in the value to return, these have the same complexity.

The exact computational complexity of Algorithm 1 can be obtained as follows. Considering that δ is usually close to 1, the computational complexity of one *SubsetSumP* call can be written as $O(I|\mathcal{T}|)$. Because we call *SubsetSumP* $(1.0 - \delta)|\mathcal{T}|$ times in Algorithm 1, the time complexity of Algorithm 1 amounts to $O(I|\mathcal{T}|^2(1-\delta))$. Although this time complexity does seem polynomial, an intrinsically infinite memory is assumed, and Algorithm 1 is pseudo-polynomial. In a readability assessment, the length of the text $|\mathcal{T}|$ is typically less than 1,000 words. In our experiments, Algorithm 1 operated in a practical manner for typical inputs.

In our application, we only need to consider integer weights because the weights are frequencies. More generally, if we also allow continuous weights, the weighted sum of independent but not identically distributed binomial distributions is called a *weighted Poisson-binomial distribution* [5]. Calculating the probability that the distribution surpasses a threshold is identical to calculating a cumulative distribution function of the distribution, and a more complicated practical algorithm was therefore proposed [16]. Because our application does not require continuous weights, however, we did not use this complicated algorithm.

5. Experiments

5.1 Comparison of Classifiers

The uncertainty-aware methods under our framework use classifiers from which we can obtain uncertainty values $p_{i,j} = P(y_{i,j} = 1)$, i.e., given learner l_j and word v_i, the probability that learner l_j knows word v_i. The classifiers can be categorized into two types based on their uncertainty values: *hard* and *soft*. The uncertainty

Algorithm 1 ProbTCSurpass: proposed algorithm for calculating the probability that a text-coverage surpasses the given δ. Here, $\times$ is a simple multiplication of two scalar values.

Input: j: Index of target learner. $\{f_1, f_2, \ldots, f_I\}$: each f_i is the frequency of word v_i in $\mathcal{T}$. $\{p_1, p_2, \ldots, p_I\}$: each p_i is the probability that learner l_j knows word v_i. δ: text-coverage threshold, $|\mathcal{T}|$: number of tokens in $\mathcal{T}$, I: number of words in $\mathcal{T}$.
Output: $p_{\text{TCsurpass}}$: Probability that the text-coverage surpasses δ

```
function ProbTCSurpass(δ)
    p_TCsurpass ← 0
    for N = ⌈|T|δ⌉ to |T| do
        p_TCsurpass ← p_TCsurpass + SubsetSumP(I + 1, N)
    end for
    return p_TCsurpass
end function
function SubsetSumP(i, N)
    if i ≤ 1 then
        if N = 0 then
            return 1
        else
            return 0
        end if
    end if
    if N ≥ f_{i−1} then
        return p_{i−1} × SubsetSumP(i − 1, N − f_{i−1})
          +(1.0 − p_{i−1}) × SubsetSumP(i − 1, N)
    else
        return (1.0 − p_i) × SubsetSumP(i − 1, N)
    end if
end function
```

values of hard classifiers are limited to 0 or 1. Those of soft classifiers are taken from $[0, 1]$.

When we have a soft classifier, we can easily obtain a hard classifier by thresholding its uncertainty values. Typically, 0.5 is used as the threshold. Formally, for a soft classifier, whose uncertainty value for learner l_j and word v_i is $p_{i,j} = P(y_{i,j} = 1|l_j, v_i)$, its corresponding hard classifier can be obtained using $\mathbf{1}\{p_{i,j} > 0.5\}$, where $\mathbf{1}\{X\}$ is the indicator function that returns 1 if X is true or 0 otherwise.

Our generalized framework is compatible with both conventional and uncertainty-aware assessment methods. In our framework, conventional uncertainty-unaware methods are expressed as methods that use only hard classifiers. By contrast, uncertainty-aware methods are methods that use soft classifiers. We compared the following classifiers.

VST is a hard vocabulary-size-based classifier conventionally used and is described in Section 2. First, it measures the learner's *vocabulary size* based on a vocabulary test. Second, according the word-frequency ranking in the British National Corpus (BNC) [3], it determines whether or not the learner knows all words whose ranks are below the vocabulary size.

LR is a logistic regression classifier that was previously used in personalized complex word identification (CWI) tasks in NLP as summarized in Section 2. The uncertainty of the personalized CWI classifiers is modeled as follows, where $\sigma(x):=1/(1 + \exp(-x))$ is the logistic function.

$$p(y_{i,j} = 1|v_i, l_j) = \sigma(\mathbf{w}_v \cdot \phi_v(v_i) - \mathbf{w}_l \cdot \phi_l(l_j)). \quad (4)$$

In Eq. (4), the input consists of *word features* $\phi_v(v_i)$ and *learner features* $\phi_l(l_j)$, where $\mathbf{w}_v$ and $\mathbf{w}_l$ are their respective weights. Word features contribute to word difficulty and are typically composed of word frequencies and word embeddings. By contrast, because the background information of a learner might be unavailable, the learner feature vector $\phi_l(l_j)$ is usually merely a *one-hot vector* of size J, i.e., a J-dimensional vector whose j-th element is 1 and all others are 0. Unlike with typical machine-learning methods, Eq. (4) allows us to interpret $\mathbf{w}_v \cdot \phi_v(v_i)$ as the difficulty of word v_i and $\phi_l \cdot \phi_l(l_j)$ as the ability of learner l_j [1], [11]. We used the **LibLinear** [13] implementation available through the **scikit-learn** toolkit.

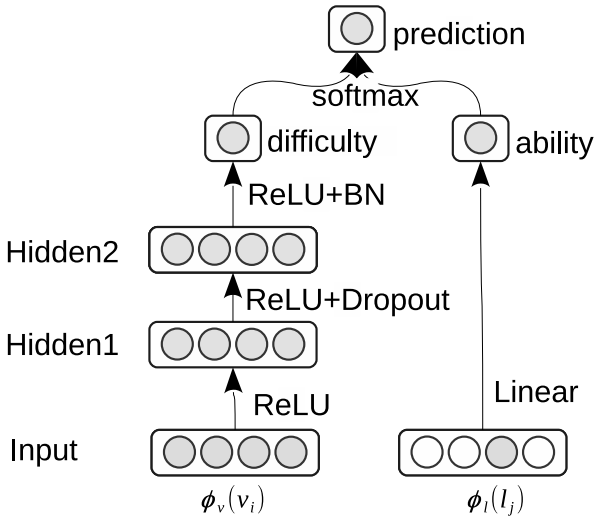


Fig. 2 NN, a neural-network-based classifier.

NN is a Neural-Net (NN) based classifier. Because NN-based classifiers have been reported to outperform logistic regression classifiers such as that used in LR, we added an NN-based classifier to our experiments. The structure of **NN** is illustrated in **Fig. 2**. The input is the same as that used in **LR**: *word features* $\phi_v(v_i)$ and *learner features* $\phi_l(l_j)$, which is merely a one-hot vector. Here, **NN** outputs uncertainty values by placing the *softmax* function in its last layer before the output. As in Fig. 2, **NN** uses typical techniques that have been reported to be effective for improving the accuracy such as dropout [30], rectified linear unit [21], and batch normalization, **BN** [17].

All hyper-parameters of **LR** and **NN** were tuned using randomly sampled validation sets taken from the vocabulary tests. To tune the hyper-parameters, we used the **optuna** toolkit [*2] with 200 iterations. **LR** has the strength of regularization and **NN** has the learning rate and the size of one hidden layer as the hyper parameters.

5.2 Dataset

As illustrated in Fig. 1, PRA methods assess whether a learner can read a given text solely based on the results of a quick vocabulary test. Therefore, to evaluate such an assessment method, we need a dataset in which each learner takes *both* vocabulary and reading comprehension tests. We created such a dataset using crowdsourcing.

We used the crowdsourcing platform Lancers, a major Japanese crowdsourcing service, of which most of the workers are native Japanese speakers. In total, 200 learners participated in building the dataset. Because a readability assessment is our

[*2] https://github.com/pfnet/optuna

focus, learners who are more motivated for reading English than listening, speaking, or writing are appropriate as participants in our experiments. Therefore, as a qualification to participate in our test, we added that they had to have taken the Test of English International Communication (TOEIC) by the English Testing Service (ETS), a widely applied test in Japan, in which quick and accurate reading is required to achieve a high score.

We used VST version A as the vocabulary test [27], which is described in Section 5.1. In addition, we used **Laufer** and **Short** as the reading comprehension tests.

Laufer includes a sample reading passage and questions [23]. This dataset was used to verify whether text coverages are effective for the PRA tasks. This verification is used for Israeli university entrance exams, and the sample was a part of one such exam. Thus, it was highly unlikely that the participants, mostly Japanese, had ever taken this test. In these exams, a test-taker who answered about 60% of their questions is assessed as capable of reading the texts in the exams. This test has a 380-word passage accompanied by 5 questions, all of which ask test-takers to select the correct answer from 4 options.

Short includes short texts also taken from the same entrance exams used by **Laufer**. In **Short**, test-takers are first presented with a short roughly sentence-length text (17.8 words on average) and asked to choose the text that is a correct paraphrase of the presented text from four choice texts.

5.3 Features for Word Difficulty

Among the compared classifiers for the PRA methods presented in Section 5.1, **LR** and **NN** can use features to estimate the word difficulty. We used the following features.

Freq is made up of word-frequency-based features from different corpora. Specifically, we used the unigram probabilities from the BNC and the unigram probabilities from the Contemporary Corpus of American English (COCA) [6]. The logarithm of the unigram probability and the raw unigram probability of both corpora were used as features.

GLOVE [29] GloVe non-contextualized word embedding: We used 50-dimensional pre-trained vectors trained from Wikipedia and the Gigaword corpus [*3]

5.4 Evaluation Measures

For the evaluation, we need to define the relationship between readability and the reading-comprehension test results. In our dataset, all reading-comprehension tests consist of a passage and questions. To correctly answer a question, a learner should be able to read both the *passage* and the *questions*. Therefore, we defined the probability that the learner can correctly answer each question as the product of the probabilities that the passage and question are readable.

To fairly compare the binary and probabilistic assessment results, we used the *mean average precision* (MAP) measure, which is typically used for this purpose, particularly in information retrieval tasks. For each learner, each readability assessment method can be viewed as retrieving questions that the learner can answer correctly from among all questions. The average precision of a method for a learner is defined as the "mean of the precision scores after each correctly answered question is retrieved." The MAP of each method is the mean of the average precision scores over all learners.

[*3] Refer to **6B** from https://nlp.stanford.edu/projects/glove/

Table 2 MAP scores of each method used for predicting the reading-comprehension test results. $\delta = 0.98$. Methods other than **Proposed** are used as the baselines.

	Methods	Laufer	Short
Conventional	VST	0.4880	0.5437
	H-LR	0.5797	0.5304
	H-NN	0.5810	0.5393
	H-LR+GLOVE	0.5113	0.5613
	H-NN+GLOVE	0.5250	0.5631
W. Avg.	A-LR	0.4880	0.4885
	A-NN	0.4880	0.4885
	A-LR+GLOVE	0.4880	0.4885
	A-NN+GLOVE	0.4880	0.4885
Proposed	UA-LR	**0.6314**	0.6533
	UA-NN	0.6172	0.6533
	UA-LR+GLOVE	0.6305	**0.6743**
	UA-NN+GLOVE	0.6159	0.6524

5.5 Compared Methods

Section 5.1 introduces hard and soft classifiers. We denote the uncertainty-aware methods obtained from the proposed framework by adding **UA-** to each classifier's name. For example, **UA-LR** denotes the uncertainty-aware method based on Eq. (2) in which a soft classifier **LR** is used as the probabilistic classifier of the words that each learner knows. As stated in Section 5.1, soft classifiers can be easily converted by thresholding their uncertainty values by 0.5. We denote the methods using the hard classifiers converted from soft classifiers by adding **H-** to each method's name. For example, **H-LR** denotes the method using the hard classifier **LR**.

We consider an easy baseline PRA method, the *weighted average*, which works by simply substituting the probabilities of knowing words weighted by frequency as the text coverage, i.e., $\sum_{i=1}^{I} f_i p_i$. We denote this case by adding **A-** to the name of each method. For example, **A-LR** denotes the results of the PRA method identically as **H-LR** except that the average probabilities using **LR** are substituted for the text-coverage.

5.6 Results

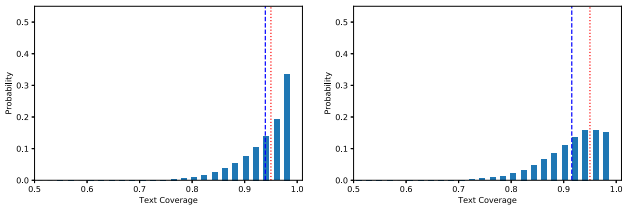
Table 2 shows the MAP values of each method. The methods in **Proposed** are uncertainty-aware methods proposed under our framework, and the other methods are baselines. All methods use the **Freq** features explained in Section 5.3. In the Method column, the names after **+** are methods with the additional word-embedding features explained in Section 5.3.

The uncertainty-aware methods consistently outperformed their corresponding conventional counterparts. For example, **UA-LR** outperformed **H-LR** and **A-LR**, and **UA-LR+GLOVE** outperformed **A-LR+GLOVE** and **UA-LR+GLOVE**. These results indicate that these methods can leverage the uncertainty values of the classifiers and produce more accurate assessments than conventional methods.

Since the weighted average underestimates the text coverage, in Table 2, the **A-** methods classified almost all cases "unread-

Method	Passage	$P(TC \geq \tau)$
H-LR	Whether people's appreciation of beauty is innate or acquired is a question addressed in ~~Plato's~~ philosophical works.	1
UA-LR	Whether people's appreciation of beauty is innate or acquired is a question addressed in ~~Plato's~~ philosophical works.	0.47

Fig. 4 A learner's assessment results. Words with a darker background are *less* likely to be known to the learner.



(a) Full train. data (b) Half train. data

Fig. 3 Text-coverage distribution of a learner.

able" when the threshold $\delta = 0.98$ was used. These results indicate that we need to find an appropriate threshold again to use the weighted average methods, which is impractical. Hence, the **A-** methods' scores are identical in Table 2.

In both **Laufer** and **Short**, the proposed uncertainty-aware methods outperformed the conventional widely used **VST** by more than 11 points. We can also see that all uncertainty-aware methods outperformed the conventional methods. This means that the gains made by leveraging uncertainty are much larger than the gains made by introducing additional features or sophisticated classifiers.

Importantly, the naive method of using the weighted average (**W. Avg.**) as a substitute for the TC in Section 5.5 achieved consistently poor results. This is because the weighted average is consistently lower than the threshold δ for the reason elaborated in Section 5.7. This result indicates that we cannot naively compare the weighted averages to the text-coverage thresholds.

The gains made with uncertainty-aware methods were much larger in **Short** than in **Laufer**. This result is intuitive because **Short** has a small number of words; therefore, the effect of leveraging the uncertainty information helps more in achieving an accurate assessment.

Regarding the cost of computation in Algorithm 1, we conducted an experiment. In the experiment, Algorithm 1 took 35.9 seconds to process 100 texts with an average of 257 words. It used 196 MiB of memory. The environment used for the experiment was Google Colaboratory, a 2-core, Intel Xeon (R) CPU @ 2.20 GHz environment with 12 GiB memory.

5.7 Qualitative Evaluation

Algorithm 1 enables us to calculate the text-coverage distribution. In **Fig. 3** (a), we show an example in which whether a learner can read **Laufer** is assessed only from the learner's vocabulary test. This learner answered all the reading comprehension questions correctly, and hence, could read the text. Each bar represents a probability that the learner's text coverage takes that value. The red dashed vertical line shows 0.95, a text-coverage threshold [*4]. Hence, the sum of the rightmost two bars is the probability that this learner is assessed as capable of reading the text. The blue dotted vertical line shows the weighted average, or the average of probabilities weighted by frequencies.

In Fig. 3 (a), the weighted average typically underestimates the text coverage, since it takes all bars into account while only the rightmost two bars are the key to assess. This makes it difficult to meaningfully compare the weighted averages with the text-coverage threshold. In this example, the weighted average is 0.94, below the threshold. The sum of the rightmost two bars is 0.52.

When we have less data, classifiers' uncertainty values tend to be ambiguous, i.e., distributions become visually flat with increased variance. Figure 3 (b) shows the results obtained under the same setting as that of Fig. 3 (a) except that the training data for classifiers or the size of the vocabulary test for each learner, was halved. We can see that the flatter distribution Fig. 3 (b) lowers the weighted average below 0.95. By contrast, the rightmost two bars show that the learner can read the text with a probability of about 30%.

Thus, assessing readability using only weighted average values is difficult since only the rightmost bars are the key. Unlike the weighted averages, when using uncertainty-aware methods based on our framework, we can use an off-the-shelf δ value, such as 0.98, and do not need to search for other threshold values. Moreover, our uncertainty-aware approach is beneficial in that the flat-shaped distribution implies that we need more data for a learner.

If we use less training data, i.e., the smaller vocabulary test, typically, but not always, the uncertainty values become ambiguous. Figure 3 (b) shows the text-coverage distribution for the same learner with Fig. 3 (a). The difference is that the classifier of Fig. 3 (b) was trained using half of the training data used for Fig. 3 (a). We can easily see that the distribution of Fig. 3 (b) is flatter than that of Fig. 3 (a), implying more training data or vocabulary questions are needed to make a more reliable assessment. In this way, unlike previous methods, our uncertainty-aware methods are useful in that these also show how reliable their readability assessments are given the limited amount of information available.

Figure 4 shows an example of the assessment results from **Short**. The darkness of the background color of each word indicates the probability that the learner does *not* know the word. **H-NN** and **UA-NN** were compared. The rightmost column shows the probability that TC surpassed δ, which was set to $\delta = 0.98$. The word "Plato" was removed from the assessment because it is a proper noun.

Figure 4 explains why the derived methods consistently outperformed the conventional approaches as shown in Table 2. Because the text in **Short** consists of only 16 words, the learner needs to know all the words. With **H-LR**, all words were classified as known to the learner because of the hard classifica-

[*4] 0.95 was chosen to make Fig. 3 easy to understand.

tion. Therefore, the text was assessed as readable to this learner. Within **UA-LR**, however, all words are lightly colored, meaning that all words, including seemingly easy words had some uncertainty in the learner's awareness of them. Specifically, words with a dark background may be unknown to the learner at a probability of 15% or more. Using Algorithm 1, **UA-LR** assessed that the text coverage of this learner surpassed δ with a probability of 0.47. Since the learner chose an incorrect paraphrase in the questions of this text, the learner might possibly have been unable to read it.

5.8 Evaluation using Easy Texts

Finally, we evaluated our method by using the publicly available dataset of easy texts. To this end, we used the OneStopEnglish corpus [31], which consists of texts annotated with readability labels, namely elementary, intermediate, and advanced. According to the paper [31], the annotation was conducted by professional native English teachers.

We used the 189 elementary texts in the dataset for the experiments. When the learner of Fig. 3 reads texts, our method predicted that the learner could read 167 texts in the 189 texts, which amounts to 88.3%. This result shows that the assessments by our method do not contradict the annotation of a publicly available readability dataset.

6. Conclusion

We propose a novel uncertainty-aware framework for the PRA task of quickly assessing whether a second language learner can read a target text using only a small vocabulary test result. Our framework can derive methods that are uncertainty-aware by leveraging the underlying classifiers' uncertainty values while guaranteeing the validity of using previously well-studied readability criteria or the text-coverage threshold values. We propose an algorithm that allows the use of such uncertainty-aware methods within the framework to make it computationally feasible. The experiment results indicate that our methods consistently outperformed the baselines in terms of accuracy. Future work includes personalized text retrieval for language learners.

Acknowledgments This paper is the revised journal version based on the conference paper [8]. This work was supported by JST ACT-I Grant Number JPMJPR18U8, ACT-X Grant Number JPMJAX2006, and JSPS KAKENHI Grant Number 18K18118, Japan. We used the ABCI infrastructure by AIST for computational resources. We appreciate anonymous reviewers for their valuable comments.

References

[1] Baker, F.B. and Kim, S.-H.: *Item Response Theory: Parameter Estimation Techniques*, Marcel Dekker (2004).
[2] Beinborn, L.: Predicting and Manipulating the Difficulty of Text-Completion Exercises for Language Learning, PhD Thesis, Technische Universität Darmstadt (2016).
[3] BNC Consortium, T.: *The British National Corpus, version 3* (*BNC XML Edition*) (2007).
[4] Chunga, T., Leongb, M.K., Looa, J. and Sia, Q.: Adaptive placement test for assessing reading comprehension levels of students studying Chinese as a second language in Singapore, *Proc. ICLC*, p.114 (2013).
[5] Daskalakis, C., Diakonikolas, I. and Servedio, R.A.: Learning poisson binomial distributions, *Algorithmica*, Vol.72, No.1, pp.316–357 (2015).
[6] Davies, M.: The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights, *Intl. J. Corp. Ling.*, Vol.14, No.2, pp.159–190 (2009).
[7] Ehara, Y.: Building an English Vocabulary Knowledge Dataset of Japanese English-as-a-Second-Language Learners Using Crowdsourcing, *Proc. LREC* (2018).
[8] Ehara, Y.: Uncertainty-Aware Personalized Readability Assessments for Second Language Learners, *Proc. 2019 18th IEEE International Conference On Machine Learning And Applications* (*ICMLA*), pp.1909–1916 (online), DOI: 10.1109/ICMLA.2019.00307 (2019).
[9] Ehara, Y., Baba, Y., Utiyama, M. and Sumita, E.: Assessing Translation Ability through Vocabulary Ability Assessment, *Proc. IJCAI*, pp.3712–3718 (2016).
[10] Ehara, Y., Miyao, Y., Oiwa, H., Sato, I. and Nakagawa, H.: Formalizing Word Sampling for Vocabulary Prediction as Graph-based Active Learning, *Proc. EMNLP* (2014).
[11] Ehara, Y., Sato, I., Oiwa, H. and Nakagawa, H.: Mining words in the minds of second language learners: Learner-specific word difficulty, *Proc. COLING* (2012).
[12] Ehara, Y., Shimizu, N., Ninomiya, T. and Nakagawa, H.: Personalized Reading Support for Second-Language Web Documents, *ACM TIST*, Vol.4, No.2 (2013).
[13] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. and Lin, C.-J.: LIBLINEAR: A library for large linear classification, *JMLR*, Vol.9, pp.1871–1874 (2008).
[14] Gordon, G., Spaulding, S., Westlund, J.K., Lee, J.J., Plummer, L., Martinez, M., Das, M. and Breazeal, C.: Affective Personalization of a Social Robot Tutor for Children's Second Language Skills, *Proc. AAAI* (2016).
[15] Hirsh, D., Nation, P., et al.: What vocabulary size is needed to read unsimplified texts for pleasure?, *Reading in a Foreign Language*, Vol.8, pp.689–689 (1992).
[16] Hong, Y.: On computing the distribution function for the Poisson binomial distribution, *Comp. Stat. & Data Anal.*, Vol.59, pp.41–51 (2013).
[17] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *Proc. ICML* (2015).
[18] Jiang, Z., Gu, Q., Yin, Y. and Chen, D.: Enriching Word Embeddings with Domain Knowledge for Readability Assessment, *Proc. COLING* (2018).
[19] Keith, K. and O'Connor, B.: Uncertainty-aware generative models for inferring document class prevalence, *Proc. EMNLP*, Brussels, Belgium (2018).
[20] Kleinberg, J. and Tardos, E.: *Algorithm design*, Pearson Education India (2006).
[21] Krizhevsky, A., Sutskever, I. and Hinton, G.E.: Imagenet classification with deep convolutional neural networks, *Proc. NIPS*, pp.1097–1105 (2012).
[22] Laufer, B.: What percentage of text-lexis is essential for comprehension, *Special Language: From Humans Thinking to Thinking Machines*, pp.316–323 (1989).
[23] Laufer, B. and Ravenhorst-Kalovski, G.C.: Lexical Threshold Revisited: Lexical Text Coverage, Learners' Vocabulary Size and Reading Comprehension, *Reading in a Foreign Language*, Vol.22, No.1, pp.15–30 (2010).
[24] Lee, J. and Yeung, C.Y.: Personalizing Lexical Simplification, *Proc. COLING* (2018).
[25] Matsushita, T.: In What Order Should Learners Learn Japanese Vocabulary? A Corpus-based Approach (2012).
[26] Nation, P.: How large a vocabulary is needed for reading and listening?, Vol.63, No.1, pp.59–82 (2006).
[27] Nation, P. and Beglar, D.: A vocabulary size test, Vol.31, No.7, pp.9–13 (2007).
[28] Paetzold, G. and Specia, L.: SemEval 2016 Task 11: Complex Word Identification, *Proc. SemEval*, pp.560–569 (2016).
[29] Pennington, J., Socher, R. and Manning, C.: GloVe: Global vectors for word representation, *Proc. EMNLP*, pp.1532–1543 (2014).
[30] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting, *JMLR*, Vol.15, No.1, pp.1929–1958 (2014).
[31] Vajjala, S. and Lučić, I.: OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification, *Proc. 13th Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, Louisiana, Association for Computational Linguistics, pp.297–304 (online), DOI: 10.18653/v1/W18-0535 (2018).
[32] Vajjala Balakrishna, S.: Analyzing Text Complexity and Text Simplification: Connecting Linguistics, Processing and Edu. Applications, PhD Thesis, Univ. Tubingen (2015).
[33] Yeung, C.Y. and Lee, J.: Personalized Text Retrieval for Learners of Chinese as a Foreign Language, *Proc. COLING*, pp.3448–3455

(2018).
[34] Yimam, S.M., Biemann, C., Malmasi, S., Paetzold, G.H., Specia, L., Štajner, S., Tack, A. and Zampieri, M.: A report on the complex word identification shared task 2018, *Proc. BEA* (2018).

Appendix

A.1 Python Code for Algorithm 1

```
class SRR:
  def SRR_(self,  i, j):
    if (i,j) in self.cache:
      return self.cache[(i,j)]
    a = self.a
    p = self.p
    if i<=0:
      return 1 if j==0 else 0
    if j>=a[i-1]:
      res =  p[i-1]*(self.SRR_(i-1,j-a[i-1])) \
      + (1.0-p[i-1])*self.SRR_(i-1,j)
    else:
      res = (1.0-p[i-1])*self.SRR_(i-1,j)
    self.cache[(i,j)] = res
    return res
  def run(self):
    a = self.a
    SUMA = sum(a)
    Tint = int(SUMA*self.T)
    ls = [ (self.SRR_(len(a),j),j) for j \
           in range(Tint, SUMA+1)]
    return sum([x[0] for x in ls])
  def __init__(self, a, p, T=0.95):
    self.a = a
    self.p = p
    self.T = T
    self.cache = {}
print(SRR([4,3],[0.3,0.7], T=0.95).run())
```

$[4, 3]$ denotes the frequency of two words. $[0.3, 0.7]$ denotes the probability of knowing the two words. This returns 0.21, the probability that the learner knows all words in this case.

A.2 Datasets

We plan to make the dataset used in this paper publicly available at http://yoehara.com/ or http://readability.jp/. Some previous datasets such as Ref. [7] are available at the former website.

Yo Ehara received his Ph.D. (Information Science and Technology) from the University of Tokyo in 2013. He is currently a full-time lecturer in Tokyo Gakugei University. His studies were partially supported by JST ACT-I and ACT-X grants. Previously, he was a full-time lecturer at the Shizuoka Institute of Science and Technology (SIST), a researcher at the Artificial Intelligence Research Center (AIRC) of the National Institute of Advanced Industrial Science and Technology (AIST), a project assistant professor at Tokyo Metropolitan University (TMU), a researcher at the National Institute of Information and Communications Technology (NICT), and a researcher at the National Institute of Informatics (NII). His research interests include educational natural language processing (NLP), item response theory, contextualized word embeddings, machine learning, and artificial intelligence. He is a member of ANLP, JSAI, IPSJ, JSiSE, ACL, and ACM.