Electronic Preprint for Journal of Information Processing Vol.26

Regular Paper

Mining Words in the Minds of Second Language Learners for Learner-specific Word Difficulty

Yo Ehara[1,a)] Issei Sato[2,b)] Hidekazu Oiwa[3,c)] Hiroshi Nakagawa[4,d)]

Received: June 11, 2017, Accepted: December 8, 2017

Abstract: While there have been many studies on measuring the size of learners' vocabulary or the vocabulary they should learn, there have been few studies on what kind of words learners think that they know. Therefore, we investigated theoretically and practically important models for predicting second language learners' vocabulary and propose another model for this vocabulary prediction task. With the current models, the same word difficulty measure is shared by all learners. This is unrealistic because some learners have special interests. A learner interested in music may know special music-related terms regardless of their difficulty. To solve this problem, our model can define a learner-specific word difficulty measure. Our model is also an extension of these current models in the sense that these models are special cases of our model. In a qualitative evaluation, we defined a measure for how learner-specific a word is. Interestingly, the word with the highest learner-specificity was "twitter." Although "twitter" is a difficult English word, some low-ability learners presumably knew this word through the famous micro-blogging service. Our qualitative evaluation successfully extracted such interesting and suggestive examples. Our model achieved an accuracy competitive with the current models.

Keywords: Rasch model, vocabulary, second language learners

1. Introduction

When learning second languages, vocabulary knowledge is as important as, or sometimes more important, than grammar. The importance of vocabulary knowledge has been a main focus in the last decade in the field of second language acquisition (SLA).

Studies regarding vocabulary knowledge of second language learners have been mainly focusing on two major tasks: devising methods for measuring the size of the second language vocabulary of learners for testing purposes [20], [24], [29] and determining the words that the learners *should* learn [25]. However, there have been few studies on what kind of words learners think that they know. Under testing environments in which learners are not motivated to exaggerate their vocabulary size, the words that learners think they know are highly likely to be actually known by the learners as discussed in Section 9. Since learners usually do not feel necessary to learn the words that they think they know again, predicting such words is beneficial for supporting the learners educationally. This is the basic research question for our research.

To study what words second language learners think they know, we focused on the *vocabulary prediction* task. In this task, we aim to build a model that predicts, given a word and a learner, whether or not the learner responds that he/she knows the word. As far as we know, Ehara et al. [8] is the only study that dealt directly with the vocabulary prediction task. They applied this task to a reading support user interface for second language learners that automatically identifies the words unfamiliar to the learner on a Web page.

The vocabulary prediction task is important for both theory and application. From the theoretical point of view, this task is interesting in that it mines the words second language learners responds to during vocabulary tests and creates a model on what kinds of words learners actually think that they know. From the model, we can interpret the patterns or tendency of the learners' process of memorizing second language words. Studying the vocabulary prediction task may also lead to determining if learners learn words that SLA experts recommend.

From the application point of view, this task can be used in user-adaptation for reading and writing applications to support second language learners. The model by Ehara et al. [8] is of this type. They successfully showed the effectiveness of their system. With the increase in Web-based language learning environments, possible data sources for learners' vocabulary knowledge are also increasing. Studying the vocabulary prediction task can shed light on these data sources, and they can be used to further understand the vocabulary knowledge of second language learners.

By using machine learning terminology, the vocabulary prediction task can be categorized as a binary classification task: given a word and a learner, it predicts whether or not the learner think that he/she knows the word. Therefore, a number of machine learning

[1] Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Koto, Tokyo 135–0064, Japan
[2] Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo, Tokyo 113–0033, Japan
[3] Recruit Institute of Technology, Chiyoda, Tokyo 100–6640, Japan
[4] Information Technology Center, The University of Tokyo, Bunkyo, Tokyo 113–8658, Japan
[a)] y-ehara@aist.go.jp
[b)] sato@k.u-tokyo.ac.jp
[c)] hidekazu.oiwa@gmail.com
[d)] nakagawa@dl.itc.u-tokyo.ac.jp

methods, such as a support vector machine (SVM) for the binary classification task, can be used as predictors. However, to answer our research question, we want predictors to be able to do more than just predict. Rather, we want predictors that are practical and useful for analysis. Specifically, we list the following properties we want predictors to have.

interpretable weight vector Most predictors use weight vectors trained with data. Weight vectors of some models can be interpreted as quantitative measures of the word difficulty and the learner ability. Interpretable weight vectors are essential for analysis to find the patterns or tendency of learners' process of memorization, and to further understand the basic research question: what kind of words do second language learners think that they know?

out-of-sample Settings in the vocabulary prediction task can be divided into two for handling new words: *in-matrix* and *out-of-sample*. The in-matrix setting does NOT support new words, i.e., there is at least one training dataset for all the words appearing in the test data. This can be seen as filling in the blanks of a learner-word matrix. In contrast, the *out-of-sample* setting support new words, i.e., some or all words in the test data are missing in the training data. To create the training data, we need to ask learners whether or not they think they know the words. Thus, creation of the training data is very financially costly and burdensome for learners. In a realistic setting, we can ask learners about only a small subset of words, and the predictors usually have to predict all the rest. The out-of-sample setting is more difficult but more realistic than the in-matrix setting.

learner-specific word difficulty This is the core beneficial property of the proposed model. Some interpretable weight vectors can determine word difficulty. However, the perceived difficulty of a word differs from learner to learner. For example, a learner interested in music may know music-related words that even high-level learners may not be familiar with. For another example, suppose that normally difficult words are used in the names of well known commercial products and services. In this case, again, low-ability learners may know these words through the product names. Thus, it is preferable for a model to be able to detect this kind of learner specialty.

Table 1 summarizes the models explained in this paper. We can see that only the proposed model supports all the properties. Although ordinary binary classifiers, such as SVMs, can be used for the vocabulary prediction task, their weight vectors cannot be used to determine word difficulty and learner ability that we want for analysis. Thus, we ruled out typical binary classifiers.

Table 1 Properties of models. The proposed model supports all preferred properties. Ordinary binary classifiers only can classify: their weight vectors are not interpretable as *word difficulty* and *learner ability* as those of the other models listed here.

	weight vector is interpretable	out-of-sample	learner-specific word difficulty
Rasch	✓	-	-
Ehara et al. [8]	✓	✓	-
Proposed	✓	✓	✓

The structure of this paper is as follows. We first focus on extending the basic interpretable model: the Rasch model [2], [27]. Although the Rasch model lacks many of the preferred properties, it provides a rough idea for the vocabulary prediction task. To explain why the Rasch model lacks many of these properties, we then introduce *the general form* of the likelihood of the Rasch model. This generalization provides a way of supporting the preferred properties. Through this generalization, we can derive the Rasch model, the model proposed by Ehara et al. [8], and the proposed model.

The contributions of this paper are as follows:

- We introduce the general form of likelihood of the Rasch model that can explain the reason this model lacks the desired properties.
- We propose a model that supports all desired properties using this general form.
- In an evaluation, our model successfully detected the specialties of second language learners, which the current models cannot detect.

2. Problem Setting

Let U be a set of learners, and V be a set of vocabulary. We denote the number of learners as $|U|$ and the number of words as $|V|$. A datum can be expressed using the triplet (y, u, v). Here, $y \in \{0, 1\}$ is the label denoting whether or not learner u responds that he/she knows word v, $(1, u, v)$ means that learner u responds that he/she knows word v, and $(0, u, v)$ means that he/she does not responds positively to word v. Using these notations, the vocabulary prediction task is defined to predict the label y given (u, v). We denote a dataset of N data as $\mathcal{D} = \{(y_1, u_1, v_1), \ldots, (y_N, u_N, v_N)\}$.

For simplicity, we assume that for one learner $u \in U$ and word $v \in V$ pair, there exists only one label y. This restriction enables us to depict the data set in a matrix form, as shown in **Fig. 1**. The rows of the matrix correspond to learners and the columns of the matrix correspond to words. Under this assumption, for one row (learner) and one column (word), there is only one cell; thus, only one label y. With this restriction, N is the number of cells in the matrix.

The dataset we used in the evaluation agrees with this restriction; however, we cannot always assume this restriction in a realistic setting. This is the reason we did not directly jump to matrix-based prediction methods such as low-rank approximation using singular value decomposition. For example, in a realistic dataset, such as word-click logs in a reading support system, contradiction and repetition are common. For contradiction, if both $(1, u, v)$ and $(0, u, v)$ appear in the dataset, it may mean these two datasets are unreliable. Repetition of multiple $(1, u, v)$ may mean that learner

Fig. 1 Two problem settings; (a) in-matrix, (b) out-of-sample.

u is more familiar with word v than just one $(1, u, v)$. All the models that we explain in the later sections of this paper can handle these cases.

Figure 1 explains the *in-matrix* and *out-of-sample* settings. The hashed areas denote the training data, and the blank areas denote the test data. In the *in-matrix* setting, the test data are randomly placed in the matrix.

3. Rasch Model

Although the vocabulary prediction task is quite novel, there have been a substantial amount of work in SLA about which words a learner *should* learn first. Many studies recommend learners to learn words according to word frequency in general corpora because word frequency can be used as a rough measure of word difficulty. Of course, the learner does not necessarily learn the words in this recommended order. As stated in the introduction, it is one of our research questions to check if learners *actually* learn in this order.

Still, we can come up with the idea that the difficulty of words determines the learners' knowledge of second language words. This idea leads to a very simple model of vocabulary prediction shown in **Fig. 2**. With this model, we predict a learner's vocabulary with the following steps:

(1) We rank words according to a measure of word difficulty.
(2) We decide the threshold for a learner.
(3) Words with greater difficulty than the threshold are predicted to be unfamiliar to the learner, and vice versa.

Although this model seems too simple, it is the core idea of the Rasch model, which has been widely used in language testing.

Given learner u and word v, the Rasch model models the probability of learner u knowing word v as follows:

$$P(y = 1|u, v) = \sigma(a_u - d_v), \tag{1}$$

where $\sigma(t) = (1 + \exp(-t))^{-1}$ denotes the logistic sigmoid function. There are two kinds of parameters to be trained:

d_v the difficulty of word v,
a_u the ability of learner u.

In the Rasch model, the subtraction of two parameters $a_u - d_v$ in Eq. (1) denotes exactly the same mechanism as the simple vocabulary prediction in Fig. 2. Here, d_v maps each word v into a point on the axis, and a_u works as a threshold. When $P(y = 1|u, v) \geq 0.5$, we can assume learner u responds that he/she knows word v. Due to the logistic sigmoid function, $P(y = 1|u, v) \geq 0.5$ holds true if and only if $a_u - d_v \geq 0$, that is, $a_u \geq d_v$. Therefore, the Rasch model determines that learner u responds that he/she knows all words whose word difficulty d_v is lower than the learners' ability a_u. Note that not only the ability of learner a_u but also the difficulty of word d_v is estimated from the data in the Rasch model.

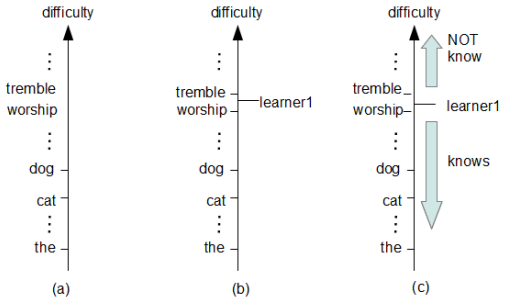


Fig. 2 Simple vocabulary prediction model. (a) First, assume there is a difficulty measure that maps each word to a point on the axis of the measure. (b) Second, each learner's ability is also mapped to a point on the same axis. (c) Third, the words with the greater difficulty compared to the point designating the learner's ability is predicted to be unfamiliar to the learner, and vice versa.

The priors for the parameters are usually set as follows:

$$P(a_u|\eta_a) = \mathcal{N}\left(0, \eta_a^{-1}\right) \quad (\forall u \in U), \tag{2}$$
$$P(d_v|\eta_d) = \mathcal{N}\left(0, \eta_d^{-1}\right) \quad (\forall v \in V), \tag{3}$$

where $\mathcal{N}$ denotes the probability distribution function of the normal distribution. Frequently, the hyper parameters η_a and η_d are set as $\eta_a = \eta_d$. If $\eta_a = \eta_d$, the parameters, d_v and a_u of the Rasch model can be obtained using a standard log-linear model solver.

One of the notable problems with the Rasch model is that it does not take into account the *out-of-sample* setting. That is, it cannot predict words that do not appear in the training set. For example, if there is a new word in a document in a reading support system, we need to re-create the training set with the new word for the system to be able to predict that word as well. This restriction makes the application systems using the vocabulary prediction task impractical.

4. General form of Likelihood

In the previous section, we stated that the Rasch model does work under the out-of-sample setting, which frequently occurs in a realistic setting. This section attempts to locate the fundamental reason the out-of-sample problem arises by generalizing the likelihood of the Rasch model.

Let us discuss the difficulty parameter d_v of the Rasch model from another perspective. If we define a function as $f(v) = d_v$, we can understand that d_v is a function that takes word v as its argument and returns the difficulty of word v. This means that we do not need to allocate the number of variables $|V|$ to determine the difficulty of a word as the Rasch model does. Instead, all that we need is a function that returns word difficulty for given word v.

We can further extend f to be the form $f(u, v)$: a function that takes learner u and word v as its argument and returns the difficulty of word v for learner u. By using $f(u, v)$, we can generalize the likelihood function of the Rasch model as follows:

$$P(y = 1|u, v) = \sigma(a_u - f(u, v)). \tag{4}$$

The Rasch model is a special version of Eq. (4) where we set $f(u, v) = d_v$. We can see that the fundamental cause of the out-of-sample problem in the Rasch model comes from this poorly designed f. There is a 1-to-1 mapping between parameters and words in this design of f. Therefore, if some words are missing in the training set, parameters arise that are not trained.

Note that Eq. (4) generalizes only the likelihood of the Rasch model. Of course, to fully define a model, we must define priors as well. Moreover, the priors must be designed carefully; otherwise, a model can produce poor results regardless of the design

Table 2 Summary of models explained so far. The Rasch model is a special case of the shared difficulty model, and the shared difficulty model is a special case of the proposed model.

Name	Design of f	Priors	Notes
Rasch	$f(u,v) = d_v$	$P(a_u\|\eta_a) = \mathcal{N}\left(0, \eta_a^{-1}\right)$ $P(d_v\|\eta_d) = \mathcal{N}\left(0, \eta_d^{-1}\right)$	-
Shared difficulty model [8]	$f(u,v) = \mathbf{w}^\top \phi(v)$	$P(a_u\|\eta_a) = \mathcal{N}\left(0, \eta_a^{-1}\right)$ $P(\mathbf{w}\|\eta_w) = \mathcal{N}\left(\mathbf{0}, \eta_w^{-1} I\right)$	Reduced to Rasch model if $\phi(v)$ is $\|V\|$-dimensional and $\phi(v) = \mathbf{1}$.
Proposed	$f(u,v) = \mathbf{w}_u^\top \phi(v)$	$P(a_u\|\eta_a) = \mathcal{N}\left(0, \eta_a^{-1}\right)$ $P(\mathbf{w}_0) = \mathcal{N}\left(\mathbf{0}, \eta_w^{-1} I\right)$ $P(\mathbf{w}_u\|\mathbf{w}_0) = \mathcal{N}\left(\mathbf{w}_0, \lambda^{-1} I\right)$	Reduced to the shared difficulty model if we set $\mathbf{w}_u = \mathbf{w_0}$ $(\forall u \in U)$.

of f.

One may think of extending the learner ability parameter a_u to be a function as well. Of course, we can do this extension in theory. However, unlike word difficulty parameters, little information is practically available for learners. Therefore, it is preferable for a model to require as little information from learners as possible. Since the complex design of f may require much information, we kept the learner ability parameter a_u simple.

4.1 Shared Difficulty Model

By redesigning f in the general form of likelihood, we can cope with the out-of-sample setting. One way to design f to be able to do this is to set it as $f(u,v) = \mathbf{w}^\top \phi(v)$. Here, $\phi : V \to \mathcal{R}^K$ is a feature function. Given word v, it returns a feature vector for it. Let K be the dimension of the feature space. Typically, frequencies from large corpora can be used as features.

Even if there is a new word in the test data and there are words in the training data that share the same features with the new word, the word difficulty of the new word can be obtained by calculating $\mathbf{w}^\top \phi(v)$. The full form of the likelihood becomes the following.

$$P(y = 1|u, v; \mathbf{w}) = \sigma\left(a_u - \mathbf{w}^\top \phi(v)\right). \tag{5}$$

Priors for the likelihood (5) are set as follows. We call this model the shared difficulty model.

$$P(a_u|\eta_a) = \mathcal{N}\left(0, \eta_a^{-1}\right) \quad (\forall u \in U), \tag{6}$$

$$P(\mathbf{w}|\eta_w) = \mathcal{N}\left(\mathbf{0}, \eta_w^{-1} I\right), \tag{7}$$

where I denotes the $K \times K$-sized identity matrix. If we set $\eta_w = \eta_a$, this model reduces to a simple l2-norm-regularized logistic regression as Ehara et al. [8] used. However, they did not mention the out-of-sample setting or the general likelihood.

5. Proposed Model

One problem in both the Rasch and shared difficulty models is that all learners share a single word difficulty measure. This means that the same ranking of a word is shared by all the learners, e.g., the word "tremble" is more difficult than *worship* according to all the learners. Thus, the Rasch and shared difficulty models cannot take into account a leaner's specialty.

In reality, it is common that even low-ability learners know difficult words with the help of their interests in a specific topic. For example, learners who are interested in music are likely to have a large vocabulary of music-related words in second languages regardless of the difficulty of the words. Modeling this kind of learner specialty is essential in designing user-adaptive supports for second language learners.

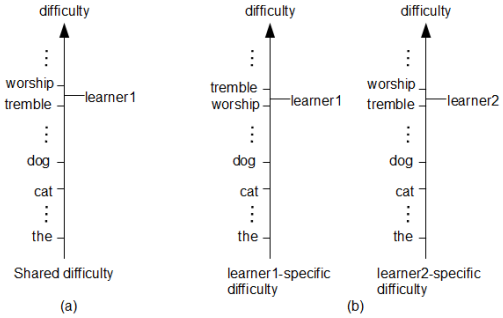


Fig. 3 Learner-specific word difficulty.

Figure 3 illustrates the difference between the shared word difficulty and the learner-specific word difficulty. On the left side of the difficulty axis, words are plotted according to the difficulty. On the right side of the axis, learner thresholds are plotted according to the learners' ability parameters a_u. The predictor determines that a learner does not know all the words above his/her threshold. In Fig. 3 (a), all three learners share the same word difficulty. Therefore, the model cannot represent a learner who responds positively to the word "worship" but does not responds positively to the word "tremble." This problem can be solved by introducing a difficulty axis for every learner as Fig. 3 (b) does. In (b), "learner 1" is modeled as knowing the word "worship" but not the word "tremble," while "learner 2" is modeled as knowing the word "tremble" but not the word "worship." This kind of flexible modeling is impossible in the Rasch and shared difficulty models.

With the general model explained above, we can easily explain the fundamental cause of this problem: in the current models, $f(u,v)$ depends only on v, and does not depend on u. Therefore, tackling this problem is simple: let $f(u,v)$ depend on u as well. In the proposed model, we define $f(u,v) = \mathbf{w}_u^\top \phi(v)$. The full form of the likelihood is shown as follows.

$$P(y = 1|u, v; \mathbf{w}_u) = \sigma\left(a_u - \mathbf{w}_u^\top \phi(v)\right). \tag{8}$$

This likelihood has far more parameters to be trained than the

current models. Since the dimension size of the feature space is K, $\mathbf{w}_u$ is a K-dimension vector. Since we have $|U|$ learners, we have $K|U|$ parameters to tune in total. Priors must be carefully designed to tune this large number of parameters. We designed the priors as follows:

$$P(a_u|\eta_a) = \mathcal{N}\left(0, \eta_a^{-1}\right) \quad (\forall u \in U), \tag{9}$$

$$P(\mathbf{w}_0) = \mathcal{N}\left(\mathbf{0}, \eta_w^{-1} I\right), \tag{10}$$

$$P(\mathbf{w}_u|\mathbf{w}_0) = \mathcal{N}\left(\mathbf{w}_0, \lambda^{-1} I\right). \tag{11}$$

Equation (11) is an important prior that does not appear in the current models. This prior makes $\mathbf{w}_u$ close to $\mathbf{w}_0$ and makes $\mathbf{w}_u$ dependent on each other. The larger the λ, the stronger this effect.

Note that both the shared difficulty model discussed by Ehara et al. [8] and the Rasch model are actually special cases of the proposed model; we *extended* the Rasch and shared difficulty models into the proposed model. The constraints to reduce the proposed model into these models are summarized in **Table 2**.

6. Estimation of Model Parameters

This section describes methods for estimating the model parameters. We use a maximum-a-posteriori (MAP) estimation for all three models: Rasch, shared difficulty, and proposed. As we explained, the shared difficulty and Rasch models are special cases of the proposed model. Therefore, we first explain the optimization of the proposed model.

The negative log of the negative log posterior of the proposed model takes the following form:

$$l(\mathbf{W}, \mathbf{a}, \mathbf{w_0}) = \sum_{i=1}^{N} nll(y_i, u_i, v_i) + \frac{\lambda}{2} \sum_{u \in U} \|\mathbf{w}_u - \mathbf{w_0}\|^2 \tag{12}$$

$$+ \frac{\eta_w}{2} \|\mathbf{w_0}\|^2 + \frac{\eta_a}{2} \sum_{u \in U} a_u^2 \ . \tag{13}$$

We define the negative log likelihood function of the proposed model as $nll(y, u, v) \stackrel{\text{def}}{=} \log(1 + \exp(-y(a_u - \mathbf{w}_u^\top \phi(v))))$. We define $\mathbf{W}$ and $\mathbf{a}$ as follows for concise notation: $\mathbf{W} = \{\mathbf{w}_u | \forall u \in U\}$, $\mathbf{a} = \{a_u | \forall u \in U\}$. This function $l(\mathbf{W}, \mathbf{a}, \mathbf{w_0})$ is convex [17] over all the variables $\mathbf{W}, \mathbf{a}, \mathbf{w_0}$. Thus, the MAP model parameters $\hat{\mathbf{W}}, \hat{\mathbf{a}}$, and $\hat{\mathbf{w_0}}$ can be estimated by minimizing $l(\mathbf{W}, \mathbf{a}, \mathbf{w_0})$ w.r.t. $\mathbf{W}$, $\mathbf{a}$, and $\mathbf{w}_0$.

Based on Kajino et al. [17], we minimize $l(\mathbf{W}, \mathbf{a}, \mathbf{w}_0)$ iteratively as follows:

minimizing w.r.t. W, a We fix $\mathbf{w}_0$ and minimize $l(\mathbf{W}, \mathbf{a}, \mathbf{w}_0)$ w.r.t. $\mathbf{W}$ and $\mathbf{a}$. Kajino et al. [17] used the Newton method for this optimization. Using the Newton method requires $O(K^2)$ memory, where K is the dimension of $\mathbf{w}_u$ and $\mathbf{w}_0$. This is problematic when K increases. To tackle this problem, we used L-BFGS [21], which requires only $O(K)$ memory, for this optimization instead. Specifically, we used the library liblbfgs [26].

minimizing w.r.t. $\mathbf{w}_0$ We fix $\mathbf{W}$ and $\mathbf{a}$ to minimize l w.r.t. $\mathbf{w}_0$. This minimization can be achieved analytically as follows:

$$\mathbf{w}_0 = \frac{\lambda}{\eta_w + |U|\lambda} \sum_{u \in U} \mathbf{w}_u \ . \tag{14}$$

We repeated these two minimizations iteratively until convergence.

Both the Rasch and shared difficulty models are special cases of the proposed model when $\mathbf{w}_u = \mathbf{w_0}$ $(\forall u \in U)$. This means that the second minimization is unnecessary for the Rasch and shared difficulty models. Thus, the parameters, i.e., the weight vector, of the Rasch and shared difficulty models can be obtained by simply performing the first minimization.

7. Evaluation

7.1 Dataset

We used the same dataset as Ehara et al. [8] used. The dataset was created in Japan in January 2009. Sixteen English as a second language learners participated in the creation of this dataset. Most were graduate students of the University of Tokyo, and Japanese was the native language of most of them.

This dataset was designed to be quite exhaustive. Every learner was handed a randomly sorted questionnaire comprising 12,000 words and asked to answer how well he/she knew the words in the questionnaire based on a five-point scale. We regarded level 5 as only $y = 1$; the learner responds positively the word. Otherwise we regarded $y = 0$; the learner does not know the word. Out of the 12,000 words, 1 word was a pseudo-word, i.e., it looks like an English word but actually is not.

Fifteen learners were paid, and 1 learner was not. Since we found that the unpaid learner's data were too noisy, we used only the data of the 15 paid learners. We had $|V| = 11{,}999$ words $\times$ $|U| = 15$ learners; 179,985 data points in total.

The negative log of the 1-gram probabilities of each word in each corpus is used as features for training. The collected corpora for feature sources are compiled in **Table 3**. Ehara et al. [8] used one large corpus, Google-1gram. However, from the perspective of SLA, it is typically not justified because it is not a general corpus; thus, its frequencies could be biased. To avoid being biased, we collected many general corpora and used them as features.

When training, hyper parameters were chosen by grid search

Table 3 Feature sources.

Corpus name	Type of English	Size (in token)	Description
British National Corpus (BNC) [30]	British	100 mil.	General corpus
The Corpus of Contemporary American English (COCA) [5]	American	450 mil.	General corpus
Open American National Corpus (OANC) [16]	American	14 mil.	General corpus
Brown corpus [14]	American	1 mil.	General corpus
Google 1-gram [3]	Mixed	1,024,948 mil.	Huge, but not general

Table 4 Top 30 words with the largest variances $Var(v)$ in descending order. Large $Var(v)$ suggests large learner-specificity. Japanese is the native language (L1) of this dataset.

$Var(v)$	word	presumed cause of learner-specificity
0.993	twitter	product name
0.886	waltz	topic specific: music, loanword in L1
0.849	kindle	product name
0.833	rink	homophone in L1 with "link"
0.827	launder	loanword in L1
0.825	bass	topic specific: music
0.823	ultraviolet	topic specific: cosmetics
0.818	chime	topic specific: music
0.804	asphalt	loanword in L1
0.802	harry	homophone in L1 with "hurry"
0.793	wooded	-
0.776	mantle	loanword in L1
0.767	trombone	loanword in L1
0.766	modulate	topic specific: computer programming
0.763	homeroom	loanword in L1
0.760	harness	-
0.760	bog	-
0.755	hearth	confused with "health"
0.750	convent	-
0.748	hurdle	loanword in L1
0.733	parson	homophone in L1 with "person"
0.732	vector	loanword in L1
0.731	haven	homophone in L1 with "heaven"
0.719	gadget	loanword in L1
0.714	lizard	-
0.713	smelt	homonym in English: past particle of "smell"
0.709	shin	homophone in L1 with "sin"
0.708	placebo	loanword in L1
0.707	lagoon	-
0.702	aha	-

and 5-fold cross validation within the training set. The set of hyper parameters that performed best in this cross validation was selected. Then, we trained the model with all the training sets using the selected hyper parameters. We then applied the model to the test set to obtain the results. For the Rasch and shared difficulty models, each hyper parameter, η_d, η_a, and η_w, was chosen by grid search from $\{0.01, 2^{-3}, 2^{-2}, 2^{-1}, 1.0, 2^1, 2^2, 2^4\}$. For the proposed model, each hyper parameter, η_a, η_w, and λ, was chosen by grid search from $\{2^{-2}, 2^{-1}, 1.0, 2^1, 2^2\}$.

7.2 Evaluation of Learner-specificity

Unlike the current models, the proposed model was designed to support learner-specific word difficulty. It is interesting to see which words are the most learner-specific.

For a measure of learner-specificity, we introduce the *variance* of learner-specific word difficulty. In the proposed model, the learner-specific difficulty $f(u, v)$ of word v for learner u is defined as $f(u, v) = \mathbf{w}_u^\top \boldsymbol{\phi}(v)$. Unlike the current models that assign single word difficulty for all learners, we can naturally define the variance of word difficulty over learners. Given the set of estimated weight vectors for all $|U|$ learners, $\{\hat{\mathbf{w}}_u \mid u \in U\}$, for word $v \in V$, we define $Mean(v)$ and $Var(v)$ as follows:

$$Mean(v) \stackrel{\text{def}}{=} \frac{1}{|U|} \sum_{u \in U} f(u, v) = \frac{1}{|U|} \sum_{u \in U} \hat{\mathbf{w}}_u^\top \boldsymbol{\phi}(v), \tag{15}$$

$$\begin{aligned} Var(v) &\stackrel{\text{def}}{=} \frac{1}{|U|} \sum_{u \in U} (f(u, v) - Mean(v))^2 \\ &= \frac{1}{|U|} \sum_{u \in U} \left(\hat{\mathbf{w}}_u^\top \boldsymbol{\phi}(v) - Mean(v)\right)^2. \end{aligned} \tag{16}$$

Table 4 lists the words with largest variances $Var(v)$ in descending order. $Var(v)$ increases when some low-ability learners know the words and some high-ability learners do not. In other words, it increases when low-ability learners know the word for some reason other than the easiness of the word, and vice versa. Table 4 is constructed from the weight vectors of the proposed model. The weight vectors are trained in the in-matrix setting. Out of 179,985 data points, 177,985 were used for training. Features and hyper parameter tuning are explained in Section 7.1. 2,000 data points were used to check the accuracy, which was 83.40%.

For example, it is very interesting and noteworthy that the word "twitter" comes at the top of the list of Table 4. This is presumably due to the famous micro-blogging service, Twitter. The word "twitter" itself is a rare word. For example, in the British National Corpus, the frequency of the word "twitter" is merely 17 while the word "the" is 6,043,900. The words whose frequency is the same with the word "twitter" are: "abet," "beguile," and "coddle." Since these three words are in the dataset as well, the rareness of words only cannot explain the large variance of the word "twitter." This dataset was created in Japan in January 2009 when Twitter was not as predominant as it is today. Therefore, some low-level learners knew the word "twitter" through the name of the service while some high-level learners did not. Additionally, Table 4 ranks another similar example at the third: "kindle." The first Amazon Kindle was released in the United States in 2007.

Likewise, we annotated presumable reasons $Var(v)$ increased in the rightmost column of Table 4. Although these reasons are speculation, it is difficult to find the correct reason learners know

a word, even for learners themselves, because we usually do not remember how we learned foreign words. Our speculations are intuitive and understandable for Japanese-native English as a Second Language (ESL) learners.

Product name The words "twitter" and "kindle" correspond to this case. When a difficult word is used as the name of a famous product, it is possible that even low-ability learners would know the word through the name of the product, which makes the variance larger.

Loanwords in L1 Some words in the second language are borrowed by the learners' native language, or L1, i.e., *loanwords*. However, the spelling of loanwords in L1 can differ from its original. For example, in the case of the word "mantle," the corresponding loanword in Japanese, the native language for most of learners of the dataset used, is spelled as "mantoru." Therefore, the difficulty has little influence on whether or not learners know the word in this case. Rather, whether or not the learner can perceive the loanword in spite of the spelling difference has more influence. Thus, even low-ability learners can perceive the meaning of the word through its corresponding loanword in L1, which makes the variance larger.

Homophones in L1 If there are two words that are homophones in the learners' native language, and one of the two words is easier than the other, a low-ability learner may mistake the difficult one for the easy one. For example, a large variance of the word "rink" is caused by low-ability learners' mistake for the word "link" because the Japanese language does not distinguish "l" and "r." For example, Japanese has no distinction between "par" and "per," the large variance of the word "parson" is presumably due to some learners mistaking this word for the word "person."

Topic specific Low-ability learners interested in a topic are likely to know the words of that topic regardless of the words' difficulty.

Homonyms in English "smelt" is a verb that means extracting metals by heat. Yet, it is also the past participle of the word "smell." Although the conjugated forms were removed from this dataset, some low-ability learners presumably did not notice it and thought that they were asked if they knew the word "smelt" as the past participle of the word "smell." Some high-ability learners presumably knew that the word "smelt" has a meaning other than the past participle of "smell" and were not asked about "smelt" as the past participle. If they did not know what was the meaning other than the past participle of "smell," they answered no in the dataset.

Note that the variance of the learners' response y for a word in the raw data *cannot* produce an interesting listing as in Table 4 because y is binary, 0 or 1. It trivially lists words of which half the learners in the dataset know. For example, if there are 15 learners in a data set, it is trivial to determine the words with the highest variance of y as those that 8 learners knew and 7 learners did not, or 7 learners knew and 8 learners did not. This means that many words have the highest y variance. In this dataset, 1,408 of 11,999 words had the highest y variance. Therefore, y variance does not produce any interesting results.

In contrast to Table 4, the words with the smallest $Var(v)$ are trivial. They are words all the learners knew or all the learners did not know. The 30 words with the smallest variances were: am, beach, doll, during, eastern, equal, excellent, green, handwriting, hungry, important, logic, love, luck, marine, paradise, shop, technical, writing, pet, unknown, loose, maker, acquittal, arduous, cot, exchequer, hindsight, innuendo, and purr.

Finally, we investigated the accuracy in the out-of-sample setting. We split the 11,999 words into 2,000 words for the test set and the rest for the training set. The size of the training data was 149,985 and the size of the test data was 30,000. The hyper parameter tuning and the feature set were the same as we stated in Section 7.1. The Rasch model achieved 66.32%, the shared difficulty model [8] achieved 77.67%, and the proposed model achieved 77.81%.

8. Related Work

The proposed model is mathematically very similar to those proposed by Evgeniou et al. [10] and Kajino et al. [17]. However, these models are for totally different purposes than ours: Evgeniou and Pontil [10] aimed at multi-task learning and Kajino et al. [17] aimed at crowd-sourcing. As the Rasch model is rarely used for these purposes, they did not mention the relationship between the Rasch and proposed models, let alone the generalization of the likelihood of the Rasch model. Strictly speaking, these two models differ from our model in that they do not include the Rasch and shared difficulty models Ehara et al. [8] as special cases while our proposed model does. Ehara et al. [8] is later published as a journal version [9].

While they also published some conference papers regarding vocabulary prediction tasks, their study has different purpose and methodology from what we proposed in this paper. Unlike our method where we uses the results of vocabulary tests, Ref. [7] proposes a graph-based model for designing tests that are used for vocabulary prediction. In machine-learning terms, in Ref. [7], they studied a graph-based active learning method and how it relates to a customary method used in the educational community. In Ref. [6], they proposed a method to evaluate translators' translation ability using vocabulary ability whereas we do not deal with translators in this paper.

We extended the word difficulty to the learner-specific word difficulty by focusing on the analysis of the vocabulary knowledge of adult second language learners. Aside from second languages, the study of vocabulary knowledge is also important for the analysis of child development in terms of native language. In computational linguistics, Kireyev and Landauer [19] proposed an extension of word difficulty called "word maturity" by focusing on the analysis of child development in terms of native language. Their extension was aimed at "track the degree of knowledge of each word at different stages of language learning" using latent semantic analysis. Thus, both their purpose and method of extending the word difficulty differ from ours.

While few have studied the vocabulary prediction task, the prediction of text readability has been of the great focus [12], [15], [18] in computational linguistics. The relationship between the

vocabulary knowledge and text readability has been thoroughly studied by educational experts [25].

A substantial amount of work has been done by mainly SLA experts in estimating the vocabulary *size*. Two major testing approaches have been proposed: *multiple-choice*, [24], and *Yes/No* [22]. For *Yes/No* tests, Eyckmans [11] studied the validity and the relation to readability prediction.

In the field of psychology, the shared difficulty model [8] is almost mathematically identical to the linear logistic test model (LLTM) [13]. Also, the vocabulary that humans memorize is studied as "mental lexicon" [1], although most of the mental-lexicon work is not aimed at predicting the vocabulary.

9. Discussion

Testing methodologies for language learners were comprehensively studied and summarized in Refs. [25], [28]. Since scoring descriptive tests becomes too costly and burdensome for large vocabularies, it is sensible to use non-descriptive tests for vocabulary prediction. There are two types of non-descriptive tests: multiple-choice tests, where test takers select the correct meaning of each word from multiple options, and self-report tests, where test takers report how well he/she knows the word. Both tests can be context-dependent, where the word is embedded within a sentence so that the word meaning can be uniquely identified, or context-independent, where words are not embedded within a sentence. To ensure that learners' responses matched what they actually knew as closely as possible, we adopted the latter type for the following reasons.

Multiple-choice tests are not always optimal for assessing learners' vocabularies [25]. First, the results are strongly influenced by the design of the incorrect options, or distractors: if the distractors can be distinguished from the correct answer too easily, the question can be answered correctly without actually "knowing" the word. Thus, a correct response to a multiple-choice question does not directly imply that the test-taker "knows" the word: he/she may have guessed the correct answer without knowing the word, or answered correctly just by chance.

Multiple-choice tests also impose a heavy burden on test-takers: if the number of words in the test is large, test-takers are motivated to answer either randomly or "I do not know" to most of the questions. Of course, self-report tests also have this issue, but because the burdens imposed by multiple-choice tests are heavier than those for self-report tests, learners taking multiple-choice tests are more likely to answer randomly.

Although self-report tests rarely suffer from these shortcomings, one major shortcoming of self-report tests is that they cannot accurately identify when test-takers are motivated to obtain high scores. However, when such motivations are not a concern, reports suggest that the results for self-report tests are closely correlated with those for well-designed multiple-choice questions [4], [23], [28]. This implies that, in these circumstances, learners are very likely to actually know the words they think they know.

Neither testing method is perfect, and learners can still make accidental responses. However, in our typical educational applications, such as reading support, A) learners are not motivated to exaggerate their vocabulary size because this may reduce the quality of the support they receive, and B) the system has to deal with a large vocabulary. Considering both of these points, we adopted self-report tests because they are more realistic. When building the dataset in Ref. [8], the learners were paid and so were unlikely to exaggerate their vocabulary size.

To minimize the gap between the self-report test results and the learners' actual vocabulary knowledge, in Ref. [8], test-takers could choose how confident they were in remembering words from five options, such as "I have heard the word, but cannot recall its meaning." We regard only the most confident case as indicating that the learner thinks he/she knows the word. Thus, a positive response implies that the learner is so confident that he/she does not think the word could have any other meaning.

To be precise, throughout this paper, we avoid saying "learners know words" and instead use the phrase "learners think they know words."

10. Conclusion

We proposed a model for the vocabulary prediction task. Although there have been few studies on it, it is interesting from both the theoretical and the practical points of views.

We introduced three preferred properties for predictors for this task: the *interpretable weight vector*, *out-of-sample setting*, and *learner-specific word difficulty*. Typical machine-learning classifiers, such as SVMs, lack the first property, interpretable weight vector. Although the Rasch model has this property, it lacks the latter two properties.

To understand why the Rasch model lacks the latter two properties, we introduced the general form of the Rasch model. From this general form, we derived our proposed model, which supports the latter two properties.

In the qualitative evaluation, we wanted to see which words are the most learner-specific. Therefore, we introduced the variance of learner-specific word difficulty and listed the top 30 words with largest variances. The results exhibited social aspects of the learners. For example, "twitter" and "kindle" came first and third, which suggests that some low-ability learners know these words through service and product names, although they are usually difficult English words. Note that this analysis is possible because the proposed model supports the third property, the learner-specific difficulty. Since the current models do not support this property, this analysis is impossible with these models. Moreover, the proposed model achieved the accuracy competitive with the current models under the out-of-sample setting, which is more realistic than the in-matrix setting.

Future work includes using topic models to determine learners' specialties. We also plan to introduce a sparse prior, such as the Laplace prior, instead of the Gaussian prior on the user-specific weight vector in Eq. (11) to obtain a more concise model in which the weights specific to each user only deviate from the overall weights.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 12J09575. We appreciate the reviewers' insightful comments.

References

[1] Amano, S. and Kondo, T.: Estimation of mental lexicon size with word familiarity database, *Proc. 5th International Conference on Spoken Language Processing* (*ICSLP*) (1998).

[2] Baker, F.B. and Kim, S.-H.: *Item Response Theory: Parameter Estimation Techniques*, Marcel Dekker, New York, second edition (2004).

[3] Brants, T. and Franz, A.: Web 1T 5-gram Version 1 (2006). LDC2006T13.

[4] Culligan, B.: A comparison of three test formats to assess word difficulty, *Language Testing*, Vol.32, No.4, pp.503–520 (online), DOI: 10.1177/0265532215572268 (2015).

[5] Davies, M.: N-grams data from the Corpus of Contemporary American English (COCA) (2011), available from ⟨http://www.ngrams.info⟩ (accessed 2012-06-23).

[6] Ehara, Y., Baba, Y., Utiyama, M. and Sumita, E.: Assessing Translation Ability through Vocabulary Ability Assessment, *IJCAI*, pp.3712–3718 (2016).

[7] Ehara, Y., Miyao, Y., Oiwa, H., Sato, I. and Nakagawa, H.: Formalizing Word Sampling for Vocabulary Prediction as Graph-based Active Learning, *EMNLP*, pp.1374–1384 (2014).

[8] Ehara, Y., Shimizu, N., Ninomiya, T. and Nakagawa, H.: Personalized reading support for second-language web documents by collective intelligence, *Proc. 15th International Conference on Intelligent User Interfaces* (*IUI 2010*), Hong Kong, China, pp.51–60, ACM (2010).

[9] Ehara, Y., Shimizu, N., Ninomiya, T. and Nakagawa, H.: Personalized Reading Support for Second-Language Web Documents, *ACM Trans. Intelligent Systems and Technology*, Vol.4, No.2, Article 31 (2013).

[10] Evgeniou, T. and Pontil, M.: Regularized multi–task learning, *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD*), pp.109–117, ACM (2004).

[11] Eyckmans, J.: Measuring receptive vocabulary size: Reliability and validity of the yes/no vocabulary test for French-speaking learners of Dutch, PhD Thesis, Radboud University Nijmegen (2004).

[12] Feng, L., Jansche, M., Huenerfauth, M. and Elhadad, N.: A Comparison of Features for Automatic Readability Assessment, *Proc. 23rd International Conference on Computational Linguistics* (*Coling 2010*): *Posters*, Beijing, China, Coling 2010 Organizing Committee, pp.276–284 (2010) (online), available from ⟨http://www.aclweb.org/anthology/C10-2032⟩.

[13] Fischer, G.: Logistic latent trait models with linear constraints, *Psychometrika*, Vol.48, No.1, pp.3–26 (1983).

[14] Francis, W.N. and Kucera, H.: *Brown Corpus Manual*, third edition, Brown University, Rhodes island (1979).

[15] Francois, T. and Fairon, C.: An "AI readability" Formula for French as a Foreign Language, *Proc. 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (*EMNLP-CoNLL*), Jeju Island, Korea, pp.466–477 (2012).

[16] Ide, N. and Suderman, K.: The Open American National Corpus (OANC) (2007). Corpus, available from ⟨http://www.AmericanNationalCorpus.org/OANC/⟩ (accessed 2012-10-24).

[17] Kajino, H., Tsuboi, Y. and Kashima, H.: A Convex Formulation for Learning from Crowds, *Proc. 26th Conference on Artificial Intelligence* (*AAAI*), Tronto, Ontario, Canada, pp.73–79 (2012).

[18] Kate, R., Luo, X., Patwardhan, S., Franz, M., Florian, R., Mooney, R., Roukos, S. and Welty, C.: Learning to Predict Readability using Diverse Linguistic Features, *Proc. 23rd International Conference on Computational Linguistics* (*Coling 2010*), Beijing, China, Coling 2010 Organizing Committee, pp.546–554 (2010) (online), available from ⟨http://www.aclweb.org/anthology/C10-1062⟩.

[19] Kireyev, K. and Landauer, T.K.: Word Maturity: Computational Modeling of Word Knowledge, *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (*ACL-HLT*), Portland, Oregon, USA, pp.299–308 (2011).

[20] Laufer, B. and Nation, P.: A vocabulary-size test of controlled productive ability, *Language Testing*, Vol.16, No.1, pp.33–51 (1999).

[21] Liu, D. and Nocedal, J.: On the limited memory BFGS method for large scale optimization, *Mathematical Programming*, Vol.45, No.1, pp.503–528 (1989).

[22] Meara, P. and Buxton, B.: An alternative to multiple choice vocabulary tests, *Language Testing*, Vol.4, No.2, pp.142–154 (1987).

[23] Mochida, K. and Harrington, M.: The Yes/No test as a measure of receptive vocabulary knowledge, *Language Testing*, Vol.23, No.1, pp.73–98 (online), DOI: 10.1191/0265532206lt321oa (2006).

[24] Nation, I.S.P.: *Teaching and Learning Vocabulary*, Heinle and Heinle, Boston, MA (1990).

[25] Nation, I.S.P.: How large a vocabulary is needed for reading and listening?, *Canadian Modern Language Review*, Vol.63, No.1, pp.59–82 (2006).

[26] Okazaki, N.: *libLBFGS: L-BFGS library written in C* (2007), Software available at ⟨http://www.chokkan.org/software/liblbfgs/⟩ (accessed 2012-10-24).

[27] Rasch, G.: *Probabilistic Models for Some Intelligence and Attainment Tests*, Danish Institute for Educational Research, Copenhagen (1960).

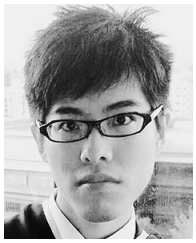
[28] Read, J.: *Assessing Vocabulary*, Cambridge University Press (2000).

[29] Schmitt, N., Schmitt, D. and Clapham, C.: Developing and exploring the behaviour of two new versions of the Vocabulary Levels Test, *Language Testing*, Vol.18, No.1, pp.55–88 (2001).

[30] The BNC Consortium: The British National Corpus, version 3 (BNC XML Edition) (2007). Distributed by Oxford University Computing Services on behalf of the BNC Consortium, available from ⟨http://www.natcorp.ox.ac.uk/⟩ (accessed 2012-10-26).

Yo Ehara received his Ph.D. (Information Science and Technology) degree from the University of Tokyo in 2013. He is currently a researcher in Artificial Intelligence Research Center (AIRC), National Institute of Advanced Industrial Science and Technology (AIST). His research interests are natural language processing (NLP) and machine learning, especially educational NLP. He is a member of ANLP, JSAI, IPSJ, and ACL.

Issei Sato received his Ph.D. degree from the University of Tokyo in 2011. He was an assistant professor at the University of Tokyo from 2011 to 2015. He is currently a lecturer at the University of Tokyo and a team leader at RIKEN AIP.



Hidekazu Oiwa received his Ph.D. (Information Science and Technology) degree from the University of Tokyo in 2015. He is currently a researcher at Recruit Holdings Co., Ltd. His research interests are machine learning and natural language processing.

Hiroshi Nakagawa was born in 1953. He received his Doctor of Engineering degree from The University of Tokyo in 1980. He joined the Information Processing Society of Japan in 1980. He is currently a professor at The University of Tokyo. His research interest is artificial intelligence and society, and privacy protection technology. He is a member of IEEE and ACM.