Predicting learner knowledge of individual words using machine learning

Drilon Avdiu, Vanessa Bui
Department of Informatics
Technical University of Munich
{drilon.avdiu,vanessa.bui}@tum.de

Klára Ptačinová Klimčíková
Class of Language Education
LMU Munich
k.klimcikova@lmu.de

Abstract

Predicting the knowledge of language learners is crucial for personalized interactions in any intelligent tutoring system for language learning. This study adopts a machine learning approach to the task of predicting the knowledge of single words for individual learners of English. We experiment with two machine learning models, neural networks and random forest, and with a set of learner-specific and word-specific features. Both the models are trained for all the learners together. However, since learner-specific features are used, the prediction is personalized for every learner. Both of the models achieve state-of-the-art results for the task of vocabulary prediction for English learners.

1 Introduction

This study is part of a larger project which attempts to develop an intelligent personal assistant for English learning called *Elia*. This assistant aims to support English learners in their informal contexts by reading or writing in English online through a browser plugin. The browser plugin also allows the collection of data about the learner's interests, knowledge and learning patterns which are used to create additional opportunities for practice in a mobile app to enhance their vocabulary acquisition. For the creation of personalized materials and personalized interaction, it is crucial to be able to automatically identify the learner's English knowledge.

Focusing on vocabulary knowledge first, the aim of this study is to create a model that would be able to predict the knowledge of single words for each learner individually. This task was firstly formulated by Ehara et al. (2014) as the *vocabulary prediction task* of which the goal is "to predict whether a learner knows a given word based on only a relatively small portion of his/her vocabulary" (p. 1374).

To tackle this problem, we adopt a machine learning approach where we engineer two sets of features, i.e., word-specific and learner-specific features, using three data sources: COCA wordlist (Davies, 2008), MRC Psycholinguistic Database: Machine Usable Dictionary. Version 2.00 (Wilson, 1988) and the English Vocabulary Knowledge Dataset (Ehara et al., 2012, 2013), the last which is also used for evaluation. We experiment with two models, i.e., Random Forest model which uses learner-specific as an input to differentiate between learners, and Neural Network model which learns the learner-specific features from the word-specific features.

The rest of the paper is structured as follows. The next section provides an overview of recent studies which are most relevant to this work (section 2). In section 3, the dataset used to evaluate the model is introduced. The features used for training and testing the models are described in section 4 and the two machine-learning models are described in section 5. Section 6 presents results and discussion. The last section summarizes the findings and suggest future directions (section 7).

2 Related Work

The knowledge prediction task is closely related to other tasks that go by different names, e.g., complex word identification (Yimam et al., 2018), automatic text simplification (Shardlow, 2014), and vocabulary size estimation (Meara and Alcoy, 2010). The studies addressing these tasks differ in their focus on a) the type of the object to be predicted, i.e., vocabulary, single words or the whole text; b) the specific aspect of the object, i.e. size, knowledge, difficulty or complexity; and c) the process name, i.e. identification, prediction, estimation. Moreover, they differ in a) the target group, i.e., native vs. non-native speakers of different languages, and b) application, i.e., reading support, vocabulary testing, text simplification. Our study focuses on the prediction of knowledge (known/unknown) of single words similar to the following studies.

Tack et al. (2016) developed an expert model which predicts known and unknown words to a learner of a given Common European Framework of Reference (CEFR) level. They annotated words and multi-word expressions in 51 texts with their level of difficulty based on a French graded vocabulary list FLELex. The same texts were then annotated by four Dutch learners of French of certain proficiency level. They used the FLELex resource, not a machine learning model, as a predictive model of the learner's lexical knowledge. They compared the predictions to learner's annotation reaching the accuracy of 87.4% to 92.3%. However, the recall of unknown words did not even reach 50%.

Alfter and Volodina (2018) is another recent study which used CEFR-annotated wordlists SVALex (François et al., 2016) and SweLLex (Volodina et al., 2016) to predict the lexical complexity (i.e. appropriate CEFR level) of single words for learners of Swedish as a second language. In addition, they used a corpus-based vocabulary list, namely the Kelly list, to extract features grouped into count-based, morphological, semantic and context-based sets. They trained several machine learning models reaching the accuracy of 59% for seen words. Features including topic distributions were found to significantly improve the accuracy.

Lee and Yeung (2018) presented a personalized complex word identification model for Chinese learners. They trained models which predict whether the learner knows a word or not for each learner separately. Graph-based active learning was used to select the most informative words which were annotated by six learners as known or unknown. They extracted several features, e.g., difficulty level, the number of characters, the word frequency in a standard and learner corpus. Trained on a set of 50 words, they obtained the best accuracy of 78% with SVM classifier with features based on word difficulty levels from pedagogical vocabulary lists.

Ehara et al. (2018) also used a personalized model trained for each learner separately. He used the dataset created by Ehara et al. (2012, 2013) where sixteen English learners annotated 12,000 words on a five-point knowledge scale making it the most exhaustive dataset for this task. For features, he used the negative log of the 1-gram probabilities of each word in several corpora. He did not use a typical machine-learning classifier because it does not have an interpretable weight vector which was the criterion of the research. Instead, he used a modified mathematical function based on the Rasch model reaching 77.8% accuracy which outperformed the other two models which were not learner-specific, namely the Rasch model and the Shared Difficulty model.

Similarly to Ehara, Yancey and Lepage (2018) learned the learners' proficiency levels and word complexities simultaneously. However, in contrast, they learned the general CEFR-level proficiency, not the learner-specific. The dataset consisted of 2,385 passages annotated by 357 learners of Korean as known or not known. For feature selection, they used Pearson's correlation and Recursive Feature Elimination with Cross Validation. With their probabilistic results, they reached the accuracy of 84.3% for unseen words at threshold 0.5.

3 Dataset

We used the dataset provided by Ehara et al. (2012, 2013) as it is the largest freely available dataset for vocabulary knowledge prediction. It contains 11,999 English single words annotated by 16 learners of English accounting for 191,984 data points in total. Most of the learners were native speakers of Japanese and attended the University of Tokyo. The sampled words were taken from the SVL 12000 wordlist (ALC, 1998). The learners were asked to indicate how well they knew the given words on a scale from 1 (I have never seen the word before) to 5 (I know the word's meaning). Similarly, as in Ehara et al. (2018) and Lee and Yeung (2018), we assigned the words marked with 5 to "known" and the rest of the words marked with 1-4 to "unknown".

4 Features

Since there can be a high variation between the knowledge of learners even of the same CEFR level (Tack et al., 2016), the goal is to make the knowledge prediction for each learner individually. As Ehara et al. (2018) rightly pointed out, “For example, a learner interested in music may know music-related words that even high-level learners may not be familiar with” (p. 801). Knowledge prediction which is learner-specific can be achieved by training an independent classifier for each learner separately (Ehara et al., 2018; Lee and Yeung, 2018). However, we train the model for all the learners together while keeping the prediction individualized. This can be achieved by adding learner-specific features which would differentiate one learner from another.

4.1 Word-specific Features

Word knowledge has often been associated with word difficulty which, in turn, has often been associated with word frequency. This was also empirically supported in the 2016 SemEval shared task for complex word identification: “word frequencies remain the most reliable predictor of word complexity” (Paetzold and Specia, 2016, p. 560). However, Tack et al. (2016) warn against word frequencies as they “approximate the use of native speakers, but do not provide any information about the frequency of words within the different stages of the L2 curriculum” (p. 230). This is, however, not the problem of frequencies but rather of the resource from which the frequencies were calculated. If the resources reflected a representative sample of the learner’s experience, whether in the classroom or beyond the classroom, word frequencies could be a reliable predictor of the knowledge of second language learners. The logic behind this is as follows: the word frequency conceptualized as the repeated opportunity to learn the word is the main predictor of the learner having learned the word. We follow this logic and create features representing different frequencies of the words taken from the Corpus of Contemporary American English (COCA) wordlist[1] (Davies, 2008) which contains word frequencies on 20,000 words from dozens of subcorpora of different genres (from academic to spoken conversations) and domains (from sports to biology). Topic distribution was also found to be the most important feature in the study by Alfter and Volodina (2018). In order to ensure comparability, the frequencies were normalized per million words across all genres.

Apart from word frequencies, we also encode the psycholinguistic properties of words into features. For this, we use the data from the MRC database[2], e.g., the number of letters, the degree of meaningfulness, the age of acquisition or the degree of abstractness. The psycholinguistic properties of words have been found to be associated with learning difficulty (Laufer, 1997), even though not directly with vocabulary knowledge. The degree of their importance in the prediction task together with the degree of importance of all the other features will be tested and described in section 4.3.

Since graded vocabulary lists have also been found to be useful in predicting the vocabulary knowledge of second language learners (Tack et al., 2016; Lee and Yeung, 2018; Alfter and Volodina, 2018), we add a feature representing CEFR difficulty level obtained from the English Vocabulary Profile (EVP)[3] resource. If one word is assigned to multiple CEFR levels, we use the lowest level of the word. If a word is not found in the database, it is automatically assigned the highest level which is the C2 level.

4.2 Learner-specific Features

For learner-specific features, we identify the number of known words in every keyword list which were created from COCA subcorpora. The proportion of known words in each keyword list should represent the knowledge of the learner across different genres and domains. The idea behind this is that if the learner knows a lot of frequent words occurring in the domain of, for example, sports, it is very probable that he/she knows another high-frequency word from this domain. However, if the learner does not know a lot of low-frequency words from the domain, it is not probable that he/she knows another low-frequency word from this domain. To operationalize this idea, we need to use a combined measure which would not

[1] Available online at `https://www.wordfrequency.info/purchase.asp`

[2] For further details, see the MRC documentation on `http://websites.psychology.uwa.edu.au/school/MRCDatabase/mrc2.html`

[3] The EVP contains information about the known words for learners of each CEFR level and is available on `https://www.englishprofile.org/wordlists`

only reflect the amount of known words but also the frequencies of the known words in a particular domain. The calculation of the learner-specific features is carried out in the following steps:

1. For each subcorpus, we extract keywords[4], that is, words which occur significantly more frequently in the specific subcorpus than in the general corpus. This results in a keyword list for each subcorpus.

2. For each keyword list, $k = \lceil \sqrt{n/2} \rceil$ of frequency bands[5] is created using the k-means algorithm[6] where n denotes the number of words in the keyword list. We use the Elkan variant of the k-means algorithm for better efficiency with a maximum number of iterations set to 300.[7]

3. In order to mitigate the effect of outliers with high frequencies, for each subcorpus, we calculate an average of the top 10 words[8] with the highest frequencies denoted as s_{max}. For the keyword lists where k is less than 10, we take k words to calculate s_{max}.

4. For each band B, we calculate the "power of band" ϕ_i by taking the difference between the subcorpus high frequency representative s_{max} and the average of all the word frequencies in the band as follows:

$$\phi_i = \frac{s_{max} - \text{avg}(B_i)}{\sum_{j=1}^{k} (s_{max} - \text{avg}(B_j))},$$

   for $i = \{1, 2, ..., k\}$.

5. For each learner, each word labeled as known from the dataset used for training/testing is looked up in the keyword lists to identify the subcorpus of the word and consequently the respective frequency band.

6. The subcorpus-specific knowledge φ_s for each learner is calculated by adding up the respective power of bands as many times as the number of words identified in those bands as follows:

$$\varphi_s = \sum_{j=1}^{k} \frac{\phi_j \cdot \mid \hat{B}_j \mid}{\mid B_j \mid},$$

   where $\hat{B}_j$ denotes the set of words which the learner knows and which belong to the band B_j, and $\big| \cdot \big|$ denotes set cardinality.

4.3 Selection of Features

The combination of the two above-mentioned types of features resulted in an exhaustive list of 105 features. Having in mind that it is very probable that the list included redundant features, a feature selection procedure was needed. To remove irrelevant and less important features, we used a Tree Classifier, a method for determining feature importance. This method gives a score for each feature where the higher the score, the more important or relevant the feature is. Not surprisingly, the word-specific features with a lot of missing values and the learner-specific features containing a limited number of keywords were ranked very low in the feature importances list and thus were discarded. Furthermore, we estimated the Pearson Correlation between the remaining features. We created groups of features with a correlation of higher than 0.99 and picked only one feature from the group with the highest rank in the feature importances list. These two procedures reduced the initial list to a final set of 39 features (see table 1 and table 2). It is worth noting that these procedures decreased the final scores slightly due to the occasional losses in information caused by the reduced word-representation.

5 Models

The objective is to train a machine learning model which would predict whether a given learner knows a given word in English or not. The problem can be formulated as follows: Let p denote the number of learners, and q the number of words in our training dataset $\mathcal{D} = \{X, Y\}$, where X denotes the set of datapoints and Y their respective labels. Let $\mathbf{u_i} = (u_{i_1}, u_{i_2}, ..., u_{i_m})^t$ and $\mathbf{w_j} = (w_{j_1}, w_{j_2}, ..., w_{j_n})^t$ denote the learner-specific features, and word-specific features, for

[4] As in Gardner and Davies (2013), we use a ratio of 1.5, i.e., all words which occur 1.5 times more often in a specific corpus compared to the general corpus are considered keywords in the specific domain.

[5] We take this number as a rule of thumb. Other heuristics can apply as well.

[6] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`

[7] We do not use a fixed number for frequency bands due to the Zipfian nature of the frequency distribution.

[8] We take this number after a manual inspection of the top frequencies in each keyword list.

Feature	
Number of letters in the word	✓
Number of syllables in the word	✓
Familiarity	✓
Concreteness	✓
Imagery	✓
Mean Colerado Meaningfulness	✓
Mean Pavio Meaningfulness	
Age of Acquisition	✓
Type	
Alphasyllable	
Status	✓
Written Capitalized	

Table 1: Initial list of features from the MRC database. Selected features for the final list are marked with a check mark.

$i = \{1, 2, ..., \mathrm{p}\}$ and $j = \{1, 2, ..., \mathrm{q}\}$, respectively. The goal is to learn the function $f : X \rightarrow Y$, or $y = f(\mathbf{w}; \mathbf{u}; \mathcal{D})$ that fits the dataset $\mathcal{D}$ to the extent of not overfitting it.

We experiment with two kinds of settings: one where both the learner-specific and word-specific features are used as input (the Complete Feature Space Dependent Model described in section 5.1) and another one where only the word-specific features are used as input and the learner-specific features are learned by the model (the Neural Network based model described in section 5.2).

5.1 The Complete Feature Space Dependent Model

For the Complete Feature Space Dependent model (CFSD), both word-specific and learner-specific features are included in the input. We tried out the following well-known machine-learning algorithms using scikit-learn (Pedregosa et al., 2011): Support Vector Machine (SVM) with various kernels, k Nearest Neighbors, Logistic Regression, and Random Forest. Random Forest (Breiman, 2001) provided the highest scores. Moreover, we have seen that training a Random Forest model, that achieved a respectable score, required way less efforts in comparison to other models. This lies to better prospects for constructing an automatic online-training pipeline in the Elia software. Consequently, Random Forest was chosen for further experimentation with the hyperparameter search.

	w	**l**
COCA total frequency	✓	✓
dispersion score	✓	
SPOKEN		
CBS (Columbia Broadcasting Company)		✓
MSNBC (Microsoft/National Brcst. Comp.)		✓
PBS (Public Broadcasting Service)		✓
NPR (National Public Radio)	✓	✓
independent	✓	✓
ABC (American Broadcasting Company)		
NBC (National Broadcasting Company)		
CNN (Cable News Network)		
FOX (Fox Broadcasting Company)		
NEWSPAPER		✓
international newspaper		✓
national newspaper		✓
local newspaper		✓
money; life		✓
miscellaneous		✓
sports; editorial		
ACADEMIC	✓	✓
education	✓	
geographical/social science		✓
law/political science; humanities		
science/technology; medicine; history		
philosophy/religion; miscellaneous		
FICTION		✓
journals	✓	✓
movies	✓	
science fiction/fantasy		
juvenile; books		
POPULAR MAGAZINES		
news/opinion		✓
religion		✓
sports; entertainment		✓
women/men	✓	
financial; science/technology		
home/health; African American		
social/arts; children		

Table 2: Initial list of features from the COCA wordlist. For word-specific features (w), the frequency of the word in the particular subcorpus was used, and for learner-specific features (l), the proportion of known words in the subcorpus was used. Selected features for the final list are marked with a check mark. Individual subcorpora are separated by a semicolon.

The Random Forest model learns the function $f : \mathbb{R}^{m+n} \to \{0, 1\}$ by using Decision Trees. The process of predicting the label for a specific input $\mathbf{x} = (\mathbf{u}, \mathbf{w})^t$ consists of all Decision Trees assigning a label. The label assigned by most of the trees is taken as the final prediction.

To come up with the best values for the parameters, we used 3-fold cross validation on 80% of the data.[9] First, we applied a random search of parameters in 100 configurations comprised of the most crucial hyperparameters of Random Forest. The selected values by random search are marked in italics in the list below. Second, we ran a grid search in a close neighborhood of the values of the parameters provided by the random search to come up with the final parameter setting. The values in the close neighborhood were chosen arbitrarily. The parameter setting that performed the best as to F1 score are marked bold:

- the use of bootstrap sampling (True, ***False***)
- the number of estimators (**55**, *75*, 95)
- the maximum depth of the trees (91, ***101***, 111)
- the minimum number of samples an internal node should contain for a split (13, ***17***, 21)
- the minimum number of samples a leaf node should contain for a split (**1**, *8*, 15)

Other preassigned parameters include the number of features to be picked randomly for a node split, which we set to the square root of the number of features, and the entropy measure which we set to Gini.

5.2 The Neural Network Based Model

In contrast to the former model, in the Neural Network based model (NN), only the word-specific features were used as input. The discrimination between the learners is achieved by constructing a unique set of parameters for each learner by the model. We learn the function $f : \mathbb{R}^n \to \{0, 1\}^p$ by a plain Fully Connected Neural Network using PyTorch (Paszke et al., 2017).

The architecture of the model is comprised of the input layer of n dimensions, 5 hidden layers with a number of nodes that changes geometrically using a factor f set to 4 (i.e., $\mathrm{f} \cdot \mathrm{n}$, $\mathrm{f} \cdot \mathrm{n} \cdot \mathrm{f}/2$, $\mathrm{f} \cdot \mathrm{n} \cdot (\mathrm{f} - 1)$, $\mathrm{f} \cdot \mathrm{n} \cdot \mathrm{f}/2$, $\mathrm{f} \cdot \mathrm{n}$) and an output layer of dimension p which is linear. For numerical stability, we use a modified binary cross-entropy loss that transforms the linear output using a Sigmoid function and afterwards employs the log-sum-exp trick.

Each of the hidden layers contains nodes with $\mathrm{ReLu}(x) = \max(0, \mathrm{x})$ activation functions.

We optimize the loss by making use of the Adam optimizer which is a more sophisticated version of the plain gradient. The hyperparameters are set upon manual analysis of the loss change. The learning rate is initially set to 0.0001, and we use mini-batches of size 15. After each layer, except layer 5, we employ a dropout regularization of 0.2 and a weight decay equal to 0.003.

We use 80% of the data for training, 3% for validation, and 17% for testing which is the same ratio as in Ehara et al. (2018). The training runs for a total of 40 epochs. For the first 20 epochs, we use the same learning rate whereas for the remaining 20 epochs we re-set the learning rate to be the $^7/_8$ of the previous value. We noticed that in this training setting, the accuracy in the validation set saturates after the 40th epoch.

6 Evaluation

6.1 Isolated Testing

We call this the isolated testing as we prevent any kind of data leakage from the training set to the testing set; the testing set is separated in the beginning before any tuning with the model is undertaken; the learner-specific features in the CFSD model are computed using information only from the words used for training. We use roughly the same ratio between the training and testing sets as Ehara et al. (2018) for comparability purposes, i.e., 80% for training in both of the models, and 20% and 17% for testing for the CFSD and the NNet model, respectively. As the classes are imbalanced—67351 labeled as 0 and 96073 as 1—we report scores other than accuracy as well (see table 3). Precision, recall, and F1 scores are calculated as a weighted average for both labels. Thus, the values of the scores are similar. The scores for both models are shown in table 3. The evaluation, including the training of our models, can be reproduced using the code accompanying this paper.[10]

[9] 3 folds were used due to limitations in time and computing capacities.

[10] The link to the code: `drive.google.com/drive/folders/1ukdm3ekkfIV_86PyGRhijC_tf07SVFxe`

	CFSD	Neural Network
Precision	**79.90%**	79.19%
Recall	**79.89%**	79.18%
F1	**79.89%**	79.18%
Cohen's Kappa	**58.26%**	56.93%
Accuracy	**79.89%**	79.18%

Table 3: The results of our models.

6.2 Discussion

The CFSD model trained on two sets of hand-crafted features, one representing the words and another representing the learners, achieved the highest accuracy, i.e., 80%. The overall results of the CFSD model support the fact that frequencies from different genres and domains—which reflect the different opportunities for learning the learner might have had—can be used as a valid representation of word-specific features. Moreover, the learner-specific features—calculated as the amount of knowledge of the keywords of specific frequencies in various genres and domains—can lead to a personalized prediction of unseen words, even in one-time training for all the learners. However, a complete feature pre-calculation, as it is the case with this model, comes with the burden of limiting the feature space to a human-defined set of features, which can not be seen as exhaustive and universal in encoding learner-specificity.

The NN model led to a slightly lower accuracy and F1 scores. This model comes with the downside of not being able to predict for learners for whom we did not train the model in the dataset as the output is fixed to the number of learners. On the other hand, it circumvents the limitation of having hand-crafted learner-specific features by learning such weight vectors from the data. In addition, we can increase the capacity of the model to encode as many learner-specific aspects as required upon data availability. Those aspects can go beyond the use of word frequencies on encoding learner-specificity as given in the CFSD model.

Comparing it to related work, both the models performed similarly to Ehara et al. (2018) who used the same dataset but different model. Their proposed model builds on top of the Rasch Model by introducing a feature map function which enriches the model with the out-of-sample setting and learner-specific learnable weight vectors. Their approach seems to be more similar to our NN model than the CFSD model in that it learns the learner representations and uses frequencies as features to represent words. However, despite the obvious similarities, there are also considerable differences, e.g., our feature map takes frequencies along dozens of different specialized corpora as opposed to few general corpora and on top adds additional non-frequency features.

Furthermore, they limit the learner-specific word difficulty vectors to the number of features constructed by their feature map which can be understood as of dimension the number of corpora they take frequencies from. On the other hand, the nature of our feature map which takes different aspects of the word into account, makes it sensical to up-project the initial feature map to higher dimensions, and thus encode learner-specificity into higher dimension weight vectors, whose size can change accordingly upon data availability.

Another difference lies in the fact that our NN approach does not model the likelihood using a single sigmoid transformation on the difference between learner's ability (a_u) and learner-specific word difficulty (w_d) and learning the parameters using a MAP estimation, but, instead, models the likelihood as a chain of ReLu transformations on standard weights. Put differently, the NN model encodes learner-specificity only on standard weights as given by the architecture. Those weights can be taken as the weights of the last hidden layer (made of $\mathrm{f} \cdot \mathrm{n}$ nodes).

7 Conclusion and Future Work

This study presented an evaluation of two supervised machine learning models which perform the task of learner's knowledge prediction of single words in the context of an intelligent tutoring system. The main challenge in this task, and thus the main goal of this study, was to make the prediction specific for every learner. We compared two approaches, one which implemented an explicit set of manually constructed learner-specific features, and another one which implemented an implicit set of learner-specific features which were learned by the model from the data.

The Random Forest model which used a complete set of hand-crafted features, both learner-specific and word-specific, led to the state-of-the-art results (accuracy of 80%) for English as a foreign language. This supports the idea of using various frequencies from different genres and domains to represent words and calculating the knowledge of keywords from those very same genre and topics to represent learners in predicting which words a given learner knows or does not know.

The Neural Network based model, using word-specific features as input and learning learner representations, led to the accuracy of 79% which sends positive signals for future work as this model does not require the construction of learner-specific features, and thus not limit the learner representation to a human-defined set of features and their calculation.

This model was initiated with the idea of building an end-to-end architecture, which firstly would encode learner specificity in the sense of dense-vector representations, and then use such encoding to create an intermediate input in concatenation with word specific features, to come up with the final prediction at the end. The idea of using the intermediate input is similar to the CFSD model, in the sense of training a one-time model which will serve our platform in long term. This way we would circumvent the limitation of our actual Neural Network based model, which does not allow the usage of a pre-trained model to generate predictions for learners whose data did not participate in the initial training. It is inferable that for such learners, we will need to run a learner-representation encoder, similar to the encoding step given in the envisioned end-to-end architecture. This is a subject of our future work.

Despite having used a large dataset of words for training and testing, the learner base was limited to 16 learners of the same language background and thus might not generalize well to heterogeneous learners which will be the case in the intelligent tutoring system Elia. However, it gives a good starting point. In future work, we plan to collect data on more learners of different background and proficiency which can be then used for further training and testing.

In conclusion, picking one model over the other introduces trade-offs, as discussed above. Thus, it is up to the designers of similar tutoring systems to decide what goes on par with their goals. For the intelligent tutoring system *Elia*, we are inclined to the idea of using a cross-learner model that exploits inter-learner similarities, such as the CFSD model, instead of using a model that does not allow for transfer of information between learners in a collaborative fashion, as the NN model. However, as stated above, our future work goes in the direction of taking the best aspects of two models. Thus, it is more likely that our platform will utilize such a model on its production state.

References

SPACE ALC. 1998. Inc. standard vocabulary list 12,000.

David Alfter and Elena Volodina. 2018. Towards single word lexical complexity prediction. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 79–88.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Mark Davies. 2008. The corpus of contemporary american english (coca): 560 million words, 1990–present. bye, brigham young university.

Yo Ehara, Yusuke Miyao, Hidekazu Oiwa, Issei Sato, and Hiroshi Nakagawa. 2014. Formalizing word sampling for vocabulary prediction as graph-based active learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1374–1384.

Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2012. Mining words in the minds of second language learners: Learner-specific word difficulty. In *Proceedings of COLING 2012*, page 799–814, Mumbai, India. The COLING 2012 Organizing Committee.

Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2018. Mining words in the minds of second language learners for learner-specific word difficulty. *Journal of Information Processing*, 26:267–275.

Yo Ehara, Nobuyuki Shimizu, Takashi Ninomiya, and Hiroshi Nakagawa. 2013. Personalized reading support for second-language web documents. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(2):31.

Thomas François, Elena Volodina, Ildikó Pilán, and Anaïs Tack. 2016. Svalex: a cefr-graded lexical resource for swedish foreign and second language learners. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 213–219.

Dee Gardner and Mark Davies. 2013. A new academic vocabulary list. *Applied linguistics*, 35(3):305–327.

Batia Laufer. 1997. What's in a word that makes it hard or easy? intralexical factors affecting the difficulty of vocabulary acquisition. *Vocabulary: Description, acquisition and pedagogy*, pages 140–155.

John Lee and Chak Yan Yeung. 2018. Automatic prediction of vocabulary knowledge for learners of chinese as a foreign language. In *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–4. IEEE.

Paul M Meara and Juan Carlos Olmos Alcoy. 2010. Words as species: An alternative approach to estimating productive vocabulary size. *Reading in a foreign Language*, 22(1):222–236.

Gustavo Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Anaïs Tack, Thomas François, Anne-Laure Ligozat, and Cédrick Fairon. 2016. Evaluating lexical simplification and vocabulary knowledge for learners of french: Possibilities of using the flelex resource. In *LREC*.

Elena Volodina, Ildikó Pilán, Lorena Llozhi, Baptiste Degryse, and Thomas François. 2016. Swellex: second language learners' productive vocabulary. In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition at SLTC, Umeå, 16th November 2016*, 130, pages 76–84. Linköping University Electronic Press.

Michael Wilson. 1988. Mrc psycholinguistic database: Machine-usable dictionary, version 2.00. *Behavior research methods, instruments, & computers*, 20(1):6–10.

Kevin Yancey and Yves Lepage. 2018. Korean l2 vocabulary prediction: Can a large annotated corpus be used to train better models for predicting unknown words? In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78.