	classification		ranking	
	ACC	PCC	pairwise ACC	PCC
native data	0.803	0.900	0.924	0.848
L2 data	0.233	0.730	0.913	0.880

Table 4: Generalization results of the classification and ranking models trained on native data applied to language testing data

Levels	1	2	3	4	5
A2	4	0	55	4	1
B1	0	0	24	6	30
B2	0	1	1	4	65
C1	0	0	0	3	64
C2	0	0	0	0	69

Table 5: Confusion matrix of the classification model on the language testing data

5.1 Generalization Experiment

First, we tested the generalization ability of the classification and ranking models trained on the WeeBit corpus on the Cambridge Exams data to see if it is possible to directly apply the models trained on native data to L2 data. Table 4 reports the results.

In the case of the multi-class classification model, the accuracy dropped greatly when the model is applied to the L2 dataset, while the correlation remained relatively high. Looking at the confusion matrix of the classifier's predictions on the L2 data (see Table 5), we notice that most of the documents in the L2 data are classified into the higher levels of WeeBit by the model. This is because, on average, the Cambridge Exams texts are more difficult than the WeeBit corpus ones which are generally targeted at children of young ages. Thus, the mismatch between the targeted levels has led to poor generalization of the classification model.

In contrast, for the ranking model, both evaluation measures are relatively unharmed when the model is applied to the L2 data. It shows that, when generalizing to an unseen dataset, the estimation produced by the ranking model is able to maintain a high pairwise accuracy and correlation with the ground truth. We believe that this is because the ranking model does not try to band the documents into one of the levels on a different basis of difficulty annotation. Instead, pairwise ranking captures the relative reading difficulty of the documents, and therefore the resulting ranked positions of the documents are closer to the ground truth compared to the classification model.

5.2 Mapping Ranking Scores to CEFR Levels

From the generalization experiment we can conclude that ranking is more accurate in predicting the CEFR levels of unseen learner texts than classification. Therefore, it is more appropriate to make use of the more informative ranking scores produced by the ranking model to learn a function that bands the scores into CEFR levels.

In learning the mapping function, we adopted a five-fold cross-validation approach. We split the Cambridge Exams dataset into five cross validation folds, with approximately equal number of documents at each level in each fold. A mapping function that converts ranking scores into CEFR levels is learnt from training folds and then tested on the validation fold in each run. The final results are averaged across the runs.

We compared three groups of methods to learn the mapping function.

(1) Regression and rounding: A regression function is learnt from the ranking scores and the ground truth labels on the training part of the dataset and then applied to the validation part. The mapped CEFR prediction is then rounded to its closest integer and clamped to range $[1, 5]$. Both linear regression and polynomial regression models are considered. The intuition behind using polynomial functions instead of a simple linear function for mapping is that the correlation of ranking scores and CEFR levels is not necessarily linear so a non-linear function might be more suitable for this task.

(2) Learning the cut-off boundary: We learn a separation boundary that bands the ranking scores to levels by maximizing the accuracy of such separation. For instance, we consider the ranked documents as a list with descending readability, with their ranking scores following the same order. If we could find a suitable cut-off boundary between each two adjacent levels in the list, then every document above the boundary would fall into the higher level, and all documents below the boundary into the lower level. In this way, the ranked documents are banded into five levels with four separation boundaries learnt.

(3) Classification on the ranking scores: The task can also be addressed as a classification problem. The ranking scores can be considered as a sin-

Mapping functions	ACC	PCC
linear regression	0.541	0.587
polynomial regression	0.586	**0.873**
cut-off boundary	0.562	0.872
logistic regression	0.610	0.862
linear SVM	**0.622**	0.864

Table 6: Results of mapping ranking scores to CEFR levels

gle dimensional feature and CEFR levels as the target value. Here, two approaches are adopted and compared, logistic regression and a linear SVM. As a matter of fact, the SVM approach can be considered as a variation of learning a separation boundary, as it tries to find an optimal decision boundary between the classes.

Table 6 shows the results of the three mapping methods. Among the three approaches for mapping ranking scores to CEFR levels (regression-based, separation boundary-based, and classification-based), the classification ones showed better results than the others in terms of accuracy. Though not as high in accuracy as the SVM, a polynomial mapping function[6] also yielded very good results in terms of PCC. Compared to the other two methods, the separation boundary-based approach performs better than a linear regression function but fails to match the polynomial regression and classification-based methods. Nonetheless, all three approaches considerably outperformed the naive generalization of the classification model from the WeeBit corpus to the Cambridge Exams data. These improvements are statistically significant at $p<0.05$ level.[7]

5.3 Domain Adaptation from Native to L2 Data

Another way to make use of the native data is to treat the task as a domain adaptation problem, where the WeeBit corpus is taken as the source domain, and the L2 data as the target domain. The idea behind this is to use out-of-domain training data to boost the performance on limited in-domain data.

EasyAdapt (Daumé III, 2007) is one of the best performing domain adaptation algorithms. It has previously been applied to essay scoring and showed good results (Phandi et al., 2015). In a two domain case, EasyAdapt expands the input feature space from $\mathbb{R}^F$ to $\mathbb{R}^{3F}$, and then applies two mapping functions $\Phi^S(\mathbf{x}) = \langle \mathbf{x}, \mathbf{x}, \mathbf{0} \rangle$ and $\Phi^T(\mathbf{x}) = \langle \mathbf{x}, \mathbf{0}, \mathbf{x} \rangle$ on source domain data and target domain data input vectors respectively. Here, $\mathbf{0} = \langle 0, ...0 \rangle \in \mathbb{R}^F$ is the zero vector. In this manner, the instance feature vectors from the WeeBit corpus and Cambridge Exams datases are augmented to three times their original dimensionality. The augmented feature space captures both general and domain specific information and is thus capable of generalizing source domain knowledge to facilitate estimation on the target domain. As there is a mismatch between the levels on native and L2 data, the pairwise ranking algorithm needs to be adapted to ensure that the preference pairs are only created from the same domain. A five-fold cross-validation is used as in previous experiments.

	pairwise ACC	PCC
EasyAdapt	0.933	0.905
native data only	0.913	0.880
L2 data only	0.943	0.913

Table 7: Results of domain adaptation from native to language testing data

Table 7 shows the results of applying EasyAdapt with the ranking model. For comparison, we also present the results obtained when we apply the model trained on the native data to the L2 data directly, and the results obtained when we train the ranking model on the L2 data only. We can see that ranking with EasyAdapt outperforms the naive generalization approach significantly ($p<0.05$), but it does not beat the results obtained when training a model on L2 data directly.

After applying the ranking model with EasyAdapt, the ranking scores can be converted to CEFR levels using the same methods as described in Section 5.2. The best mapped CEFR estimation is achieved with a linear SVM classifier on the ranking score, reaching an ACC of 0.707 and PCC of 0.899. Compared to the naive generalization of the classification model from native to L2 data, the mapped estimation is less influenced by the mismatch between difficulty levels in the two domains (see Table 8).

[6] A 4th order polynomial function is adopted because it yields better results compared to other orders.

[7] Throughout this paper, we test significance using t-test for ACC and Williams' test (Williams, 1959) for PCC.

Levels	1	2	3	4	5
A2	11	3	0	0	0
B1	2	9	0	1	0
B2	0	0	13	0	2
C1	0	0	2	9	2
C2	0	0	0	4	10

Table 8: Confusion matrix of the mapped estimation after EasyAdapt application on one of the cross-validation folds

Type	ACC	PCC
L2 data only	0.785	0.924
self-training	0.797	0.938

Table 9: Results of self-training

5.4 Using Self-training to Enhance the Classification Model

In addition to the domain adaptation, we experimented with self-training to boost the performance on the limited L2 data with the native data. To the best of our knowledge, neither of the approaches has been applied to readability assessment before.

Self-training is a commonly used semi-supervised machine learning algorithm that aims to use the large amount of unlabelled data to help build a better classifier on a small amount of labeled data (Zhu, 2005). When using native data to boost model performance on L2 data with self-training, the L2 data is regarded as labeled instances, and the native data as unlabeled ones. A model is trained on the L2 data and then used to score the native data. The most confident *K* instances as well as their labels are added to the training set. Then the model is re-trained and the procedure is repeated. A five-fold cross-validation is used in evaluation as before.

We have experimented with a grid search on *K*'s and the number of iterations, and found out that whatever the choice of the parameters is, the model performance degrades with self-training when the unlabeled instances are added blindly to all levels of the L2 dataset. Taking into account the mismatch in the difficulty levels between the native and L2 texts, we adapted the algorithm to add the unlabeled data only to the lower three levels of the L2 dataset. The best result is achieved with K=10 and 9 iterations, with 270 texts added in total (as shown in Table 9). It seems reasonable to compare the results of this approach to those obtained with a model that is trained directly on the L2 data. Hence, we include the results of this model in Table 9 for comparison.

The results show that self-training can significantly (p<0.05) help estimating readability for L2 texts by including a certain amount of unlabeled data (in this case, the native data) in training. However, the range of the reading difficulty covered by the unlabeled data may influence the model performance.

6 Conclusions and Future Work

We investigated text readability assessment for both native and L2 learners. We collected a dataset with text tailored for language learners' readability and explored methods to adapt models trained on larger existing native corpora in estimating text reading difficulty for learners. In particular, we developed a system that achieves state-of-the-art performance in readability estimation, with ACC=0.803 and PCC=0.900 on native data, and ACC=0.785 and PCC=0.924 on L2 data, using a linear SVM. We compared a ranking model against the classification model for the task and showed that although a ranking model does not necessarily outperform a classification one in readability assessment on the same data, it is more accurate when generalizing to an unseen dataset. Following this, we showed that, by applying a ranking model and then learning a mapping function, the model trained on the native data can be applied to estimate the CEFR levels of unseen text effectively. This model achieves an accuracy of 0.622 and PCC of 0.864, and considerably outperforms the naive generalization of the classification model, which achieves an accuracy of 0.233 and PCC of 0.730.

In addition, we experimented with domain adaptation and self-training approaches to make use of the more plentiful native data to produce better estimation of readability when the L2 data is limited. When treating the native data as a source domain and L2 data as a target domain, applying the EasyAdapt algorithm for ranking achieves an accuracy of 0.707 and PCC=0.899. The best result is achieved by using self-training to include native data as unlabelled data in training the classification model, with ACC=0.797 and PCC=0.938.

Future work will focus on the improvement of readability assessment framework for L2 learners and the identification of the optimal feature set that can generalize well to unseen text.