Assessing Grade Levels of Texts via Local Search over Fine-Tuned LLMs

Changfeng Yu and Jie Wang
Richer Miner School of Computer and Information Sciences, University of Massachusetts, Lowell, MA, U.S.A.

Keywords: Automatic Grade Assessment, Linguistic Features, Large Language Models, Local Search.

Abstract: The leading method for determining the grade level of a written work involves training an SVC model on hundreds of linguistic features (LFs) and a predicted grade generated by a fine-tuned large language model (FT-LLM). When applied to a diverse dataset of materials for grades 3 through 12 spanning 33 genres, however, this approach yields a poor accuracy of less than 51%. To address this issue, we devise a novel local-search algorithm called LS-LLM independent of LFs. LS-LLM employs different FT-LLMs to identify a genre, predict a genre-aware grade, and compare readability of the text to a randomly selected set of annotated works from the same genre and grade level. We demonstrate that LS-LLM significantly improves accuracy, exceeding 65%, and achieves over 92% accuracy within a one-grade error margin, making it viable for certain practical applications. To further validate its robustness, we show that LS-LLM also enhances the performance of the leading method on the WeeBit dataset used in prior research.

1 INTRODUCTION

The leading method for automatic grade assessment (Lee et al., 2021) trains a multi-label SVC model on all 255 known LFs and grade predictions from a fine-tuned BERT model, which produces the best results to date on the datasets of WeeBit (Vajjala and Meurers, 2012) and Newsela (Xia et al., 2016). WeeBit consists of texts categorized into five age groups and spans a limited range of genres, while Newsela contains only news articles. These datasets fall short of our requirements for evaluating grade levels across diverse genres of written materials.

To address this need, we collected all freely available written works from the CommonLit Digital Library (CommonLit.org) along with their genres and grade levels. This results in a dataset of 1,654 written works spanning 33 genres for U.S. students in grades 3 through 12. We refer to this dataset as CLDL1654, or simply CLDL.

Applying the leading method using the code provided by Lee et al., we train a multi-label SVC model with all 255 LFs on CLDL and grade levels predicted by FT-*M*, with *M* being, respectively, BERT, RoBERTa, BART, and GPT-4o. These models all exhibit low accuracy below 51%. We further show that using only about 10% of the LFs, varying for different LLMs, the trained SVC model can achieve accuracy levels comparable to those obtained using all 255 LFs.

This calls for a new approach independent of LFs. Initially, we attempted to fine-tune a GPT-4o classifier and use few-shot prompting with examples of texts at each grade level and genre. However, experimental results show that the accuracy of these two approaches is below the SVC-based models, which is likely due to the complexity introduced by genre variation–texts from different genres at the same grade level can vary significantly in style, structure, and vocabulary. Furthermore, a single few-shot prompt cannot capture all representative examples, and even if it could, GPT-4o may be influenced by conflicting signals across genres.

This suggests the necessity of a new way to leverage the vast knowledge depository and strong inference capability of an LLM. To this end, we devise a local-search method called LS-LLM that employs a number of FT-LLMs, each tailored to a specific task. LS-LLM falls in the framework of AI-oracle machines (Wang, 2025), which decomposes the grade assessment into sub-tasks of genre identification, grade assessing for texts of a specific genre, and readability comparison for texts in the same genre. We address each sub-task using an FT-LLM and apply a local-search algorithm to determine the appropriate grade level for a given text through an iterative process, guided by the outputs of these sub-tasks.

We show that LS-LLM consistently outperforms the leading method on CLDL and WeeBit with GPT-

Yu, C. and Wang, J.
Assessing Grade Levels of Texts via Local Search over Fine-Tuned LLMs.
DOI: 10.5220/0013674400004000

In *Proceedings of the 17th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2025) - Volume 1: KDIR*, pages 224-231

4o and freely available BERT and RoBERTa as the underlying LLMs.

This paper is organized as follows: Section 2 provides a brief overview of prior works. Section 3 evaluates the prior leading method. Sections 4 and 5 describe LS-LLM in detail and report evaluation results. Section 6 concludes the paper.

2 RELATED WORK

Early systems for automatic readability assessment include Dale-Chall (Jeanne Sternlicht Chall, 1995) and Fog (Gunning, 1969), which use linear regressions to estimate readability based on lexical features of word length, sentence length, syllable count, and word frequencies. These features, however, fall short in addressing semantics, discourse structure, and other nuanced elements of language. Feng et al. (Feng et al., 2009) analyzed a broader set of cognitively motivated features, such as the number of entities in a sentence. Tonelli et al. (Tonelli et al., 2012) reported a set of syntactic features related to part of speech, phrasal structure, and dependency structure of the text. These more complex features have been shown to correlate better with part-of-speech usage and complex nominal construction.

More sophisticated systems were later developed using machine learning techniques. For example, Schwarm (Schwarm and Ostendorf, 2005) and Ostendorf employed linguistic features (LFs) such as syntactic complexity, semantic difficulty, and discourse coherence to train an SVM model for predicting text readability. The performance of these methods depends heavily on how well the LFs capture the information related to text readability (Lu, 2010).

Lee et al. (Lee et al., 2021) presented the leading method that trains an SVC model on 255 LFs combined with a grade level of a written work predicted by an FT-PLM. SVC was chosen as the non-neural classifier as it performs well on classification with small training datasets. They evaluated their method using WeeBit (Vajjala and Meurers, 2012) and Newsela (Xia et al., 2016) as training data. Likewise, Deutsch et al. (Deutsch et al., 2020) showed that incorporating only 86 LFs into LLMs can improve the accuracy, especially with small training datasets. Recent advances in LLMs have led to interest in reliably assessing and manipulating the readability of the text, including measuring and modifying the readability of text (Trott and Rivière, 2024; Engelmann et al., 2024).

3 GRADE ASSESSING WITH LFS

LFs can be computed using the Python library at https://github.com/brucewlee/lingfeat. for any input text. We use the code provided by Lee et al. (Lee et al., 2021) to train an SVC model using all 255 LFs, employing various FT-LLMs to predict the grade level of a written work. In particular, we divide CLDL into a standard 80-20 split for training and testing, and leverage the Scikit-Learn library. All subsequent model training, fine-tuning, and evaluation will be performed using this same 80-20 split.

We fine-tune BERT, RoBERTa, BART, and GPT-4o separately so that each can assign a grade to a given written work. To fine-tune BERT, RoBERTa, and BART, we apply the 5-fold cross validation method using Hugging Face's transformers library with 10 epochs and 1 batch size. We use fastai's learn.lr_find() to find the optimal learning rate during fine-tuning. To fine-tune GPT-4o we use default settings of GTP-4o and the following prompt template (Note that in all prompts we specify that the user is an experienced assessor of the language and literature curricular for the public K-12 schools in the US):

> User: Your task is to determine the grade level of the following text. {text}
>
> Assistant: {grade level}

We name the corresponding SVC classifiers as SVC-255/M, where M represents, respectively, FT-BERT, FT-RoBERTa, FT-BART, and FT-GPT-4o. We generalize this notation to SVC-k/M to represent a model trained using k LFs with an FT-LLM M. Figure 1 depicts the fine-tuning and training processes and the application of the models.

In addition to exact matches, where the predicted grade aligns perfectly with the true grade, referred to *adjacent distance-0* (AD-0), we also include cases where the predicted grade has an error margin of one grade level, referred to as *adjacent distance-1* (AD-1) (Heilman et al., 2008). This adjustment accounts for possible inconsistencies and potential imperfections in human evaluations, providing a more nuanced assessment. Using the same notation, we can define adjacent distance-2 (AD-2) similarly.

Table 1: Evaluation of the leading method.

Model	AD-0	AD-1	AD-2
SVC-255/BERT	0.4988	0.8871	0.9153
SVC-255/RoBERTa	0.5022	**0.8915**	0.9262
SVC-255/BART	0.4932	0.8902	0.9226
SVC-255/GPT-4o	**0.5024**	0.8891	**0.9324**
FT-GPT-4o (no LFs)	0.4512	0.8611	0.8922

Table 1 shows the results on the test data of CLDL using the four SVC classifiers trained on the training

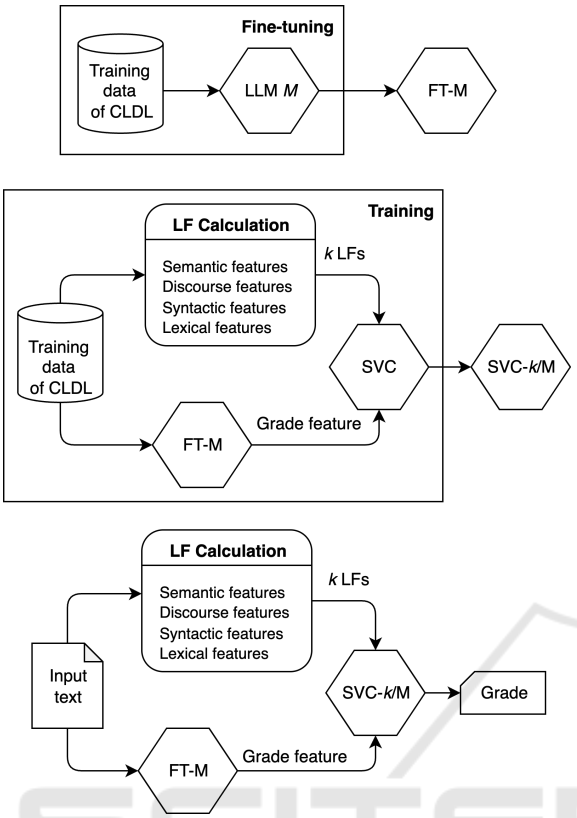

Figure 1: Schematics of training and application of the hybrid model.

data of CLDL, as well as the result generated by the fine-tuned GPT-4o without LFs, where fine-tuning is carried out using the training data of CLDL. The numbers in boldface indicate the largest in the underlying column.

It is evident that SVC-255/GPT-4o achieves the highest AD-0 accuracy. This result can be regarded as the performance ceiling when all LFs are incorporated.

We observe that among the 255 LFs, some are essential, others are redundant, and a few are even counterproductive. This observation motivates the following investigation into how many of these features are primarily responsible for the model's performance.

According to their definitions, we select 100 LFs that appear to be more significant and use them, instead of all 255 LFs, to train an SVC model with an FT-LLM for predicting grade levels using the leading method. Our results demonstrate that the SVC model using these 100 LFs achieves the same accuracy as that trained on all 255 LFs when paired with the same FT-LLM for predicting grades. These 100 LFs, along with their feature names and definitions, are available at https://github.com/readability-assessment/ARA/blob/main/LFs.pdf, classified into four categories: semantic, discourse-based, syntactic, and lexical features.

We further observe that not all these 100 LFs are necessary to achieve the same level of accuracy. To identify how many LFs from these 100 LFs are essential, we intend to carry out a grid search as follows: Enumerate all combinations of these 100 LFs, and identify the smallest number of LFs such that an SVC trained on them reaches the performance upper bound. However, this approach results in an exponential blowup, rendering it intractable to implement. Moreover, our experiments also indicate that, because different PLMs are trained differently, the essential LFs may vary across different PLMs.

To reduce computation time, instead of exhaustively evaluating all combinations of LFs, we conduct a constrained grid search as follows: (1) For each PLM, select an approximately equal number of LFs from each category independently at random, starting from 0 to max, where max is the largest number of LFs in a category, with a total number of LFs from 0 to 100. (2) Train an SVC model using these LFs in the same method as before. (3) After training the SVC classifier with the reduced number of LFs, we assess its performance on the test data of CLDL. To ensure robustness, we repeat the experiment three times for each value of k and the final accuracy reported is the average across these three runs.

Table 2 depicts the evaluation results of AD-0. It is evident that increasing the number of LFs does not necessarily lead to improved AD-0 accuracy, as some LFs can be counterproductive. For example, SVC/GPT-4o with 20 LFs achieves an AD-0 accuracy of 50.5%, which drops to 50.3% when using 24 LFs, and ultimately falls further to 50.24% when all LFs are used, as shown in Table 1, where "SVC/M with k LFs" is defined in the same manner as "SVC/M with all LFs," and k represents the number of LFs used.

4 LS-LLM

Let M be the LLM chosen for fine-tuning genre assessors, grade assessors (one for each genre), and text comparators (one for each genre).

4.1 Genre Assessor

We observe that it is more appropriate to compare readability between written works in the same genre, as texts from different genres such as poem and biography can vary significantly, even at the same grade level. To support this, we fine-tune M to create a genre assessor that predicts the genre of a given text.

Table 2: Evaluation of SVC models with k LFs

Model	AD-0 with value of k									
	0	4	8	12	16	20	24	28	32	36
SVC-k/BERT	0.432	0.463	0.487	0.489	0.496	**0.504**	0.501	0.493	0.502	0.498
SVC-k/RoBERTa	0.420	0.455	0.461	0.475	0.489	**0.506**	0.495	0.499	0.502	0.506
SVC-k/BART	0.428	0.452	0.469	0.481	0.484	0.498	**0.501**	0.499	0.501	0.493
SVC-k/GPT-4o	0.451	0.473	0.489	0.502	0.499	**0.505**	0.503	0.498	0.499	0.502

If M is a generative model such as BART and GPT-4o, we fine-tune M using the following prompt template, where the {genre list} is all genres in Table 3, contained in CLDL:

> User: Your task is to determine the genre of the following text. {text}
> The list of genres is given below: {genre list}
> Assistant:{the genre of the text}

If M is a non-generative transformer such as BERT and RoBERTa, we fine-tune M as a classifier following the standard procedure.

4.2 Partitioning

It is evident from Table 3 that texts in CLDL are unevenly distributed across genres, and for certain genres, there is an insufficient number of texts spanning all grade levels. To address these issues, we group texts by similar genres to ensure that each genre group contains an adequate number of texts at each grade level. To do so, let $E = [e_1, e_2, \ldots, e_n]$ denote the list of n genres for the underlying dataset ($n = 33$ for CLDL), sorted in descending order according to the percentage, p_i, of the number of texts with genre e_i over the total number of texts in the dataset. Let K be the smallest number such that $\sum_{i=1}^{K} p_i \geq \Delta$ for $\Delta \in (\frac{1}{2}, 1]$.

We partition E into K clusters: $C_1, C_2, \ldots, C_K$, with genre $e_i \in C_i$ for $i = 1, \ldots, K$. We call e_i the base genre of C_i. For each remaining genre of $e_{K+1}, \ldots, e_n$, we place it in C_i if it has the highest similarity with the base genre e_i of C_i among all clusters. The similarity of two genres is calculated as the cosine similarity of the BERT embeddings of sentences describing the respective genres. We generate these sentences using GPT-3.5 with the following prompt template:

> User: Your task is to generate an explanation of the genre {name of the genre} in one sentence.

Denote by D_i for $i = 1, \ldots, K$ the subset of texts and the corresponding grades whose genres are in C_i, as shown in Figure 2.

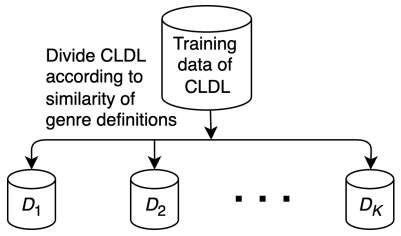

Figure 2: Grouping texts according to genre participation.

4.3 Grade Assessors

If M is a generative model, we fine-tune M to predict grade levels for written works in each subset D_i using the following prompt template, resulting in a grade assessor denoted as GA_i:

> User: Your task is to determine the grade level of the following text. {text}
> Assistant: {grade level}

If M is a non-generative model, we fine-tune M as a classifier to classify grade levels for files in D_i following the standard procedure, still denoted as GA_i.

4.4 Text Comparators

From each D_i, we select independently at random m (e.g., $m = 10$) written works at a given grade level g to create a set of reference texts, denoted by

$$F_{i,g} = \{f_{i,g,j} \mid j = 1, \ldots, m\}. \quad (1)$$

Next, we construct a labeled pairwise dataset for fine-tuning M as follows: For each grade level $g \in [g_{\min}, g_{\max} - \ell]$ with $\ell \geq 1$, where $g_{\min}$ and $g_{\max}$ denote, respectively, the lowest and the highest grade levels in the dataset (e.g., in CLDL, $g_{\min} = 3$ and $g_{\max} = 12$), let

$$P^{+}_{i,g+j} = \{((x,y), +1) \mid (x,y) \in F_{i,g} \times F_{i,g+j}\},$$
$$P^{-}_{i,g+j} = \{((x,y), -1) \mid (x,y) \in F_{i,g+j} \times F_{i,g}\},$$

where $+1$ and -1 are labels, and $1 \leq j \leq \ell$ sets the range of grade levels. Let

$$P_i = \bigcup_{g_{\min} \leq g \leq g_{\max} - \ell, 1 \leq j \leq \ell} \left(P^{+}_{i,g+j} \bigcup P^{-}_{i,g+j}\right). \quad (2)$$

Finally, if M is a generative model, we fine-tune it on P_i to create a text comparator, denoted by TC_i, with the following prompt template:

Table 3: The genres in CLDL in descending order of percentage, where "R" represents the ranking of a genre in terms of the number of texts in that genre.

R	Genre	%	R	Genre	%	R	Genre	%
1	Information text	0.3622	12	Fable	0.0160	23	Science fiction	0.0046
2	Poem	0.1709	13	Psychology	0.0153	24	Religious text	0.0038
3	Short story	0.1041	14	Fantasy	0.0122	25	Political theory	0.0038
4	Essay	0.1041	15	Folktale	0.0122	26	Allegory	0.0030
5	Fiction	0.0574	16	Opinion	0.0115	27	Autobiography	0.0030
6	Speech	0.0428	17	Myth	0.0076	28	Legal document	0.0023
7	Biography	0.0383	18	Primary source doc	0.0076	29	Satire	0.0022
8	News	0.0214	19	Historical fiction	0.0068	30	Letter	0.0015
9	Memoir	0.0176	20	Philosophy	0.0067	31	Main ideas	0.0007
10	Non-fiction	0.0161	21	Drama	0.0054	32	Magical realism	0.0007
11	Interview	0.0161	22	Historical document	0.0053	33	Skill lesson	0.0007

User: You are provided with a pair of texts delimited with XML tags. Your task is to determine which of the two texts is more difficult to read.

<text 1> $\{x_i\}$ <text 1>

<text 2> $\{y_i\}$ <text 2>

Assistant: {<text 1> or <text 2>}

If M is a non-generative model, we fine-tune M on P_i as a binary classifier to determine which of the two input texts is more difficult to read following the standard procedure. Figures 3 and 4 depict the process of fine-tuning these models.

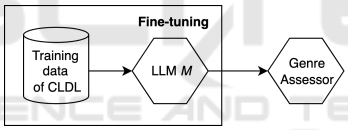

Figure 3: A schematic for fine-tuning the genre assessor.

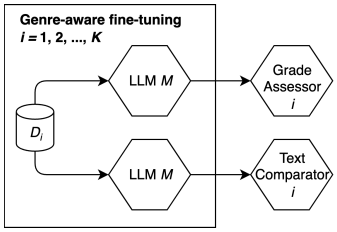

Figure 4: A schematic for fine-tuning genre-aware grade assessors and text comparators.

4.5 The Local-Search Algorithm

Let F be an input text. Figure 5 depicts the data flow of LS-LLM.

1. Use the genre assessor to predict F's genre, denoted by e.
2. Case 1: $e \in C_i$ for some i ($1 \le i \le K$).
   (a) Use the grade assessor GA_i to predict an initial grade level of F, denoted as g, and use it as the starting point for carrying out the local search.

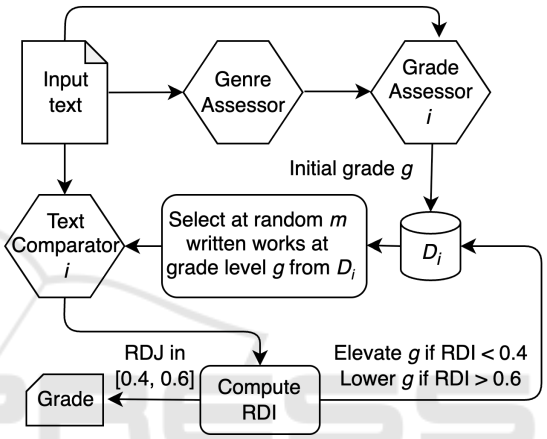

Figure 5: Data flow of LS-LLM.

   (b) Select at random m files from D_i with grade level g, denoted by $F_{i,g}$.
   (c) Use the text comparator TC_j to compare F with each file in $F_{i,g}$. Let $n^L_{i,g}$ denote the number of texts in $F_{i,g}$ with a lower grade level then g. Define the *relative difficulty index* (RDI) by

$$RDI_{i,g} = \frac{n^L_{i,g}}{m}. \quad (3)$$

   Case 1.1: $RDI_{i,g} < 0.4$. If $g > g_{\min}$, then set $g \leftarrow g-1$, one grade lower, and repeat the algorithm. Otherwise, the local search concludes with an output "F is easier than Grade $g_{\min}$."
   Case 1.2: $0.4 \le RDI_{g,i} \le 0.6$. The local search concludes with the current value of g being the final grade of F.
   Case 1.3: $RDI_{i,g} > 0.6$. If $g < g_{\max}$, then set $g \leftarrow g+1$, one grade higher, and repeat the algorithm. Otherwise, the local search concludes with an output "F is harder than Grade $g_{\max}$."
3. Case 2: $e \notin E$. Namely, e is unseen in the training data. We identify the existing genre that is most similar to e using the same clustering method applied to genres clustering and proceed as in Case

1, applying the genre-aware grade assessor and text comparator associated with that genre.

Remark. While we may randomly select reference works from the training data for a given grade on the fly, independent of those used for fine-tuning the grade comparators, our experiments show that this approach achieves almost the same accuracy.

5 EVALUATION

We first evaluate the accuracy of the genre assessor, grade assessor, and text comparator on CLDL. We then evaluate the overall performance of LS-LLM/*M* on both CLDL and WeeBit. We would like to apply LS-LLM to Newsela, but we have not received permission to access Newsela at the time of writing[1].

For brevity, we sometimes refer to a model as a *D*-based model if it is trained or fine-tuned on the dataset *D*. We carry out evaluations for WeeBit in two settings: (1) Repeat the same fine-tuning process for WeeBit as for CLDL but with five levels of readability using a genre-agnostic grade assessor. (2) Apply the CLDL-based genre assessor, genre-aware grade assessors, and genre-aware text comparators to WeeBit. Finally, we compare the performance of genre-agnostic LS-LLM with genre-aware LS-LLM on both CLDL and WeeBit, as well as the number of visits to LLMs and the actual running time.

5.1 Evaluation of CLDL-Based Models

The test sets for the genre assessor and grade assessor are the test data of CLDL. The test set for the text comparator is constructed in the same way as for constructing P_i (see Equation 2) with the following setting: For CLDL: $g_{\min} = 3$, $g_{\max} = 12$, and $\ell = 2$. For WeeBit: $g_{\min} = 1$, $g_{\max} = 5$, and $\ell = 1$, where the readability level is treated as the grade level. We use the average precision to measure accuracy. For measuring the genre assessor, a predicted genre is considered correct if it falls in the correct cluster of genres. Table 4 shows the evaluation results, where GenA stands for "genre assessor," GraA for "grade assessor," and TexC for "text comparator."

It can be seen that the CLDL-based genre assessor, genre-aware grade assessor, and genre-aware text comparator using GPT-4o achieve the highest accuracy compared to other LLMs, with accuracies exceeding 82%, 45%, and 85%, respectively. We will use the CLDL-based genre assessor using GPT-4o as the default genre assessor for its highest accuracy. It is worth noting that the genre assessor may generate a new genre not present in the training data.

[1]Access to Newsela requires permission, as does WeeBit.

Table 4: Evaluation of CLDL-based models.

Model	GenA	GraA	TexC
BERT	0.7435	0.3912	0.7692
RoBERTa	0.7847	0.3968	0.8121
BART	0.7422	0.3975	0.8010
GPT-3.5	0.7833	0.4017	0.8244
GPT-4o	**0.8206**	**0.4512**	**0.8538**

5.2 Evaluation of LS-LLM on CLDL and WeeBit

WeeBit doesn't provide genre information. To resolve this, we use the genre assessor trained on CLDL to generate genres for all 3,115 written works in WeeBit. Table 5 depicts the results. A total of 857 written works have generated genres not included in CLDL, highlighting the advantage of using a generative model over a traditional classifier.

Table 5: Statistical results of predicted genres for WeeBit, where #Art represents "the number of texts"

R	Genre	#	R	Genre	#
1	Info text	1424	14	Lang lesson	18
2	News	386	15	Short story	18
3	Advertisement	365	16	Mathematics	14
4	Interview	345	17	Poem	8
5	Information	99	18	Biography	8
6	Science	78	19	Drama	7
7	Statement	71	20	FLLR	5
8	Info technology	68	21	Religious	3
9	Summary	49	22	Recipe	2
10	Education	47	23	Joke	2
11	Literary analysis	37	24	Case study	1
12	Philosophy	30	25	Character Analysis	1
13	Opinion	29			

Selecting $\Delta = 0.65$ and 0.75, respectively, for CLDL and WeeBit yields $K = 4$ for both datasets in genre partitioning, which means that datasets are partitioned into four groups, with the top four genres in Tables 3 and 5 being, respectively, the base genres for the underlying cluster. This partition provides a sufficient number of works in each D_i spanning all grade levels. We set $m = 10$ to construct the set of reference works $F_{i,g}$ (see Equation 1) for each D_i.

Table 6 depicts the evaluation results, where GPT-4o (direct) generates grade levels using a few-shot prompt, LS-L stands for LS-LLM, /3.5 and /4o stand for /GPT-3.5 and /PGT-4o, and SVC-255/4o is trained over, respectively, CLDL and WeeBit.

It can be seen that, for both CLDL and WeeBit under both AD-0 and AD-1 accuracy, LS-LLM/*M*

Table 6: Evaluation results of various models trained or fine-tuned on their respective datasets.

Model	AD-0		AD-1	
	CLDL	WeeBit	CLDL	WeeBit
GPT-4o (Direct)	0.4378	0.7623	0.8420	0.8220
FT-GPT-4o	0.4512	0.8950	0.8611	0.9050
SVC-255/4o	0.5024	0.9187	0.8891	0.9532
LS-L/BERT	0.6387	0.9195	0.9103	0.9593
LS-L/RoBERTa	0.6516	0.9221	0.9179	0.9611
LS-L/BART	0.6425	0.9250	0.9101	0.9678
LS-L/3.5	0.6526	0.9316	0.9174	0.9668
LS-L/4o	**0.6542**	**0.9327**	**0.9202**	**0.9697**

for all M outperforms the leading method trained with all LFs, which in turn outperforms fine-tuned GPT-4o, and fine-tuned GPT-4o outperforms out-of-the-box GPT-4o. In particular, under the measure of AD-0, for CLDL, LS-LLM/GPT-4o achieves a substantial 23.20% improvement. Even the least-performant model, LS-LLM/BERT, surpasses the leading method with a notable 21.34% improvement. For WeeBit, LS-LLM/GPT-4o achieves a 1.50% improvement over the leading method.

In can also be seen that all models achieve higher accuracy on WeeBit compared to CLDL. This is likely because WeeBit features coarser readability levels, allowing certain grade predictions that are incorrect for CLDL to be correct for WeeBit.

Table 7: Evaluation of CLDL-based models on WeeBit.

Model	AD-0	AD-1
SVC-255/GPT-4o	0.4412	0.5929
LS-LLM/BERT	0.4648	0.6246
LS-LLM/RoBERTa	0.4701	0.6290
LS-LLM/BART	0.4677	0.6263
LS-LLM/GPT-3.5	0.4711	0.6302
LS-LLM/GPT-4o	**0.4716**	**0.6308**

Next, we evaluate the transferability of CLDL-based LS-LLM/M on WeeBit. For a written work F in the test set of CLDL, if LS-LLM predicts "F is easier than Grade 3," we classify F as belonging to Grade 3. Similarly, if LS-LLM predicts "F is harder than Grade 12," we classify F as belonging to Grade 12. We map the predicted grade by LS-LLM/M as follows: (1) Texts easier than Grade 3 are classified as Level 1. (2) Texts at Grades 3 and 4 are classified as Level 2. (3) Texts at Grades 5 and 6 are classified as Level 3. (4) Texts at Grades 7, 8, and 9 are classified as Level 4. (5) Texts at Grades 10, 11, 12, and those harder than Grade 12 are classified as Level 5. Table 7 presents the evaluation results, where SVC/GPT-4o with all LFs is trained on CLDL.

5.3 The Role of Genres

We compare the performance of LS-LLM/GPT-4o with genre-agnostic and genre-aware grade assessor *GA* and text comparator *TC* fine-tuned on, respectively, where Genre-agnostic models are trained without organizing the training data according to genre. Table 8 depicts the evaluation results.

Table 8: The average AD-0 accuracy of LS-LLM/GPT-4o.

Method	AD-0	
	CLDL	WeeBit
Genre-agnostic	65.22%	93.25%
Genre-aware	**65.42%**	**93.27%**

It appears that the genre-aware method performs slightly better; however, the advantage is marginal, which is somewhat counterintuitive. This may be attributed to the imbalance in the dataset across genres, where a few dominant genres disproportionately influence the results. To enable a fairer comparison, a more balanced dataset is necessary for future studies.

Table 9 (a) and (b) show, respectively, the maximum and average numbers of visits to LLMs and the running time of LS-LLMs.

Table 9: The number of visits to LLMs and running time, where G-AG and G-AW stand for, respectively, genre-agnostic and genre-aware

	Maximum		Average	
	CLDL	WeeBit	CLDL	WeeBit
G-AG	41	21	14.8	12.9
G-AW	32	22	12.6	12.5

(a)

		Worst-case time		Average time	
		CLDL	WeeBit	CLDL	WeeBit
BERT	G-AG	28.23	14.13	10.21	9.13
	G-AW	23.26	15.60	9.04	9.01
GPT-4o	G-AG	42.34	21.7	15.38	13.33
	G-AW	33.06	22.76	13.22	12.92

(b)

It can be seen that, in general, the genre-agnostic approach requires more visits to the underlying fine-tuned PLM models compared to the genre-aware approach. This is expected, as the genre-aware approach is confined to a smaller set of genres, which results in a faster local search process. Consequently, the actual running time of LS-LLM using fine-tuned PLMs like BERT or RoBERTa, which run locally, is significantly shorter compared to LS-LLM using fine-tuned commercial PLMs such as the GPT-4o API. Table 9 depicts the comparison results of running time, where the fine-tuned BERT models are run on a NVIDIA GeForce RTX 3090 GPU.

6 CONCLUSIONS

We presented a novel local search method for readability assessment, leveraging fine-tuned models over a selected PLM for various tasks. Our experiments demonstrated that the proposed local search method significantly enhances ARA accuracy over the leading method. Investigations for further improvements of accuracy can be carried out along the following lines: (1) Construct a dataset that is larger and more balanced than CLDL. Specifically, for each genre, we aim to collect a sufficient number of written works that are evenly distributed across all grade levels. This will eliminate the need to partition the dataset by similar genres and enable fairer comparisons between genre-agnostic and genre-aware grade assessment and readability evaluation methods. (2) Explore alternative black-box LLMs with improved fine-tuning capabilities to enhance the accuracy of various tasks. (3) Investigate white-box LLMs, such as the LLaMA models, to optimize fine-tuning for specific tasks.

REFERENCES

Collins-Thompson, K. (2014). Computational assessment of text readability: A survey of current and future research running title: Computational assessment of text readability.

Deutsch, T., Jasbi, M., and Shieber, S. (2020). Linguistic features for readability assessment. In Burstein, J., Kochmar, E., Leacock, C., Madnani, N., Pilán, I., Yannakoudakis, H., and Zesch, T., editors, *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17.

Engelmann, B., Kreutz, C. K., Haak, F., and Schaer, P. (2024). Arts: Assessing readability and text simplicity. In *Proceedings of EMNLP*.

Feng, L., Elhadad, N., and Huenerfauth, M. (2009). Cognitively motivated features for readability assessment. In Lascarides, A., Gardent, C., and Nivre, J., editors, *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 229–237.

Filighera, A., Steuer, T., and Rensing, C. (2019). *Automatic Text Difficulty Estimation Using Embeddings and Neural Networks*, pages 335–348.

Gunning, R. (1969). The fog index after twenty years. *Journal of Business Communication*, 6:13 – 3.

Hale, J. (2016). Information-theoretical complexity metrics. *Lang. Linguistics Compass*, 10:397–412.

Heilman, M., Collins-Thompson, K., and Eskenazi, M. (2008). An analysis of statistical models and features for reading difficulty prediction. *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*.

Holtgraves, T. (1999). Comprehending indirect replies: When and how are their conveyed meanings activated? *Journal of Memory and Language*, 41(4):519–540.

Jeanne Sternlicht Chall, E. D. (1995). *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books.

Lee, B. W., Jang, Y. S., and Lee, J. (2021). Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686. Republic.

Lee, B. W. and Lee, J. (2020). Lxper index 2.0: Improving text readability assessment for l2 English learners in South Korea.

Lu, X. (2010). Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15:474–496.

Peabody, M. A. and Schaefer, C. (2016). Towards semantic clarity in play therapy. *International Journal of Play Therapy*, 25:197–202.

Schwarm, S. and Ostendorf, M. (2005). Reading level assessment using support vector machines and statistical language models. In Knight, K., Ng, H. T., and Oflazer, K., editors, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 523–530.

Tonelli, S., Tran Manh, K., and Pianta, E. (2012). Making readability indices readable. In Williams, S., Siddharthan, A., and Nenkova, A., editors, *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 40–48. Canada.

Trott, S. and Rivière, P. (2024). Measuring and modifying the readability of English texts with GPT-4. In Shardlow, M., Saggion, H., Alva-Manchego, F., Zampieri, M., North, K., Štajner, S., and Stodden, R., editors, *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 126–134. Linguistics.

Vajjala, S. and Meurers, D. (2012). On improving the accuracy of readability classification using insights from second language acquisition. pages 163—-173.

Wang, J. (2025). Ai-oracle machines for intelligent computing. *AI Matters*, 11:8–11.

Xia, M., Kochmar, E., and Briscoe, T. (2016). Text readability assessment for second language learners. In Tetreault, J., Burstein, J., Leacock, C., and Yannakoudakis, H., editors, *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22.