Exploring hybrid approaches to readability: experiments on the complementarity between linguistic features and transformers

Rodrigo Wilkens[1], **Patrick Watrin**[1], **Rémi Cardon**[1],
Alice Pintard[1], **Isabelle Gribomont**[1,2], **Thomas François**[1]
[1]CENTAL, IL&C, University of Louvain, Belgium
[2]Royal Library of Belgium (KBR)
{rodrigo.wilkens, patrick.watrin, remi.cardon, alice.pintard,
isabelle.gribomont, thomas.francois}@uclouvain.be

Abstract

Linguistic features have a strong contribution in the context of the automatic assessment of text readability (ARA). They have been one of the anchors between the computational and theoretical models. With the development in the ARA field, the research moved to Deep Learning (DL). In an attempt to reconcile the mixed results reported in this context, we present a systematic comparison of 6 hybrid approaches along with standard Machine Learning and DL approaches, on 4 corpora (different languages and target audiences). The various experiments clearly highlighted two rather simple hybridization methods (soft label and simple concatenation). They also appear to be the most robust on smaller datasets and across various tasks and languages. This study stands out as the first to systematically compare different architectures and approaches to feature hybridization in DL, as well as comparing performance in terms of two languages and two target audiences of the text, which leads to a clearer pattern of results.

1 Introduction

A significant proportion of the population suffers from their poor reading skills in their everyday life (Schleicher, 2019, 2022), for example to access medical information (Friedman and Hoffman-Goetz, 2006) or to process administrative tasks (Kimble, 1992). This issue may be tackled with Automatic Readability Assessment (ARA); for example by automating recommendations of texts suited to specific reading levels (Pera and Ng, 2014; Sare et al., 2020).

ARA has leveraged automatic annotation of textual features, and Machine Learning (ML) algorithms. In this context, ARA has largely been modeled using feature engineering (Collins-Thompson, 2014; François, 2015; Vajjala, 2021). Current works rely on distributed representations of texts (i.e. embeddings) (Cha et al., 2017; Filighera et al., 2019) and Deep Learning (DL) (Nadeem and Ostendorf, 2018; Azpiazu and Pera, 2019; Martinc et al., 2021), yielding improvement over linguistic feature-based systems (e.g., Deutsch et al. (2020); Martinc et al. (2021) for English and Yancey et al. (2021) for French). Consequently, DL has become the new standard in ARA. However, linguistic feature engineering has not been completely discontinued (Imperial, 2021; Weiss and Meurers, 2022). We emphasize two main reasons for that. First, obtaining audience-specific data to produce large corpora, required for DL, is difficult, and vanilla transformers tend to achieve low performance on small readability datasets (Lee et al., 2021). Second, feature-based approaches bring knowledge from cognitive psychology and the modelling of difficulty (Chall and Dale, 1995), offering insights on how textual characteristics affect readers (Javourey-Drevet et al., 2022).

In this work, we focus on hybrid models as a way to combine the accuracy of DL with the grounded interpretability of features, with minimal pre-training costs.[1] We aim to identify an effective architecture for combining linguistic features and transformers for ARA, keeping in mind that there may be an overlap of the information encoded in both representations (Goldberg, 2019; Rosa and Mareček, 2019; Jawahar et al., 2019; Kim et al., 2020). Although this work focuses on ARA, the methodology presented here can be applied to other tasks, particularly those tasks that rely on a restricted data set. The main contributions of this paper are: (1) a systematic analysis of how hybrid architectures compare with traditional ones[2], (2) recommendations for the best hybrid architecture for ARA, and (3) a study of how those models are impacted by corpora properties (e.g. language, or

[1]Note that other types of hybrid models, such as multimodal models, are outside the scope of this work.

[2]Developed model is available on gitlab.com/rswilkens/linguistic-features-in-transformers.

Findings of the Association for Computational Linguistics: EACL 2024, pages 2316–2331
March 17-22, 2024

L1 vs. L2). The paper is structured as follows: we discuss existing work in more details (Section 2), we detail our approach (Section 3) and present the results we obtained (Section 4). We then present an in-depth error analysis (Section 5) before concluding (Section 6).

2 Related Work

The inclusion of linguistic features in DL models has been done in various areas of NLP. In some cases, the purpose is to provide additional information that a DL model does not have access to (e.g. information about products (Amplayo et al., 2022)). Additionally, linguistic information can be included to facilitate the learning task, by providing complementary information or information poorly presented in the model. The inclusion of features in DL requires changes in the architecture, which can be done by adding additional layers or modifying the existing ones[3]. In this section, we examine how this modification in architecture is carried out in NLP and particularly in ARA.

2.1 Hybrid Models

Feature integration methods can be divided into two categories, depending on whether integration is direct or indirect.

Direct (or explicit) integration consists in concatenating feature vectors and embedding vectors. This method is simpler to implement than the indirect method and is the most widely used. It enriches the networks' input with fine-grained linguistic information that may be under-represented or particularly important in the networks' embeddings. Balagopalan and Novikova (2020), for example, connect the last layer of BERT to a vector of 119 lexical and syntactic features to improve an Alzheimer's Disease (AD) detection system. The same method can be found in several other systems: Complex Word Identification (Ortiz-Zambrano et al., 2022); Automatic Essay Scoring (Prabhu et al., 2022); Abusive Language Detection (Koufakou et al., 2020); Natural Language Understanding (Zhang et al., 2020); and assigning a CEFR (Common European Framework of Reference) level to a text[4](Schmalz and Brutti, 2021). Peinelt et al. (2021) proposed an alternative concatenation method by injecting pre-trained (non contextual) embedding into the BERT architecture. To that end, they projected the embedding sequence to BERT's internal dimensions and squashed the output values to a range between -1 and 1.

Indirect (or implicit) integration consists in orienting fine tuning by associating one or more auxiliary tasks with the main task. For example, Zhou et al. (2019) propose a multi-task architecture which aims at simultaneously integrating morpho-syntactic (POS-tagging), syntactic (component and dependency parsing) and semantic (span and dependency semantic role labeling) information into the model.

2.2 Hybrid Models for ARA

Deutsch et al. (2020) investigated if adding linguistic-based characteristics to deep learning models can increase their performance in ARA. They compared conventional ML (SVMs, Linear Models, and Logistic Regression), CNNs, Transformer, and HANs to do this. They employed the numerical output of a neural model as a feature itself, concatenated with language data, and then fed into one of the non-neural models. Deutsch et al. (2020) identified strong differences in models ranking depending on the corpora.

Imperial (2021) advocated for concatenating raw embeddings with constructed language feature sets and feeding them to typical machine-learning techniques. Li et al. (2022) built a BERT-based model with feature projection and length-balanced loss. They derive a set of topic features by grouping related words with similar difficulty levels. To produce orthogonal features, these features are concatenated and projected (Qin et al., 2020) to the neural network features. According to Li et al. (2022)'s ablation study, the most significant improvement is related to the length-balanced loss they proposed, whereas the features had a minor impact. Lee et al. (2021) employed a soft labeling approach (i.e., the fine-tuned BERT probabilities of the prediction are concatenated with linguistic data), and used the whole to train Random Forest models. Liu and Lee (2023) compared hard labels (the fine-tuned BERT prediction is concatenated with linguistic data), following Deutsch et al. (2020), soft labels, and sentence concatenated with features embeddings, for investigating passage-level ARA. They found that Hard Labels and Soft Labels outperform transformers, but the sentence concatenated model

[3]The modification of existing layers implies the invalidation of pre-trained models, which represents a large training cost and is therefore outside the scope of this work.

[4]Direct integration has also been used with non-linguistic information: Zhang et al. (2021) and Amplayo et al. (2022) integrate extra-textual data (e.g. user or product information) in various classification contexts (mainly sentiment analysis).

performed the poorest (similarly to a vanilla transformer model).

In order to give a first indication of the performance of the different strategies for combining features with transformers, Table 1 compiles the performance of the different works presented in this section. Thus, the initial observation points to the use of soft labeling, but the number of features is different between the works and the results using concatenation are based on one corpus only.

3 Methodology

In order to find the best approach for combining features and embeddings for ARA, we carried out a systematic comparison of architectures by comparing hybrid and non-hybrid (baselines) architectures. To this end, we selected 4 readability corpora with various characteristics (Section 3.1), on which we computed linguistic features (Section 3.2) before comparing the performance of the 8 architectures described in Section 3.3. To this aim, we split each corpus into train, validation and test sets (60/20/20) using stratified cross validation with groups defined based on target difficulty level and text genre (when available). For comparing performance, we applied the Friedman and Mann-Whitney U tests.

3.1 Corpora

Assessing our architectures requires corpora in which the reading difficulty of each text has been evaluated according to a reference difficulty scale[5], In this work, we opted for 4 corpora that cover two readability tasks (one targeting native speakers and the other targeting language learners) as well as two languages (English and French)[6].

French as Native Language (FLM[7]) (Wilkens et al., 2022a) is composed of 334 text documents from Belgian school material. They are divided into 9 levels (from grade 4 to grade 12) and three domains (History, Science, and French language). The level of a text is the level of the textbook it was taken from.

French as Foreign Language (FLE[8]) (François and Fairon, 2012; Yancey et al., 2021) is composed of 2,734 text documents extracted from French as a foreign language (FFL) textbooks published between 2001 and 2018. The level of each document ranges across five CEFR levels (Council of Europe, 2001) and is the same as the textbook from which it was taken.

Cambridge (Xia et al., 2016) is a collection of 330 reading texts from the Cambridge English Exams, explicitly designed for L2 learners at different proficiency levels. The corpus is divided into five CEFR levels, depending on the proficiency levels.

Clear (Crossley et al., 2021) is a set of 4,716 excerpts (written between 1875 and 1922) scored by 1,116 teachers from CommonLit Ease according to their easiness for a student (8 to 17 y/o in the American curriculum), where the final text readability score is the probability of text easiness based on the Bradley-Terry model.

3.2 Linguistic Feature Annotation

Before comparing our different architectures, we needed to identify the relevant features for each corpus. The first challenge is to identify tools that annotate both languages in a similar way. In this sense, the FABRA toolkit (Wilkens et al., 2022a) and its English version (Wilkens et al., 2022b) are suitable options. This toolkit annotates numerous linguistic variables relevant for readability. Since many of these variables are at the word or sentence level, the toolkits use various statistical aggregators (e.g., mean, percentile and skewness) to create the features for each text aiming at a more detailed description of the linguistic variables.[9]

After the 4 corpora were annotated, we had to identify an appropriate set of features to be used in the hybrid models. To this end, we opted for the mRMR (Maximum Relevance Minimum Redundancy) method[10] (Ding and Peng, 2003). More specifically, following Zhao et al. (2019), we used the FCQ variant of mRMR (a combination of Random Forest, Randomized Dependence Coefficient, and Quotient). We explored 10 different sizes of feature sets (10, 20, 30, 40, 50, 100, 200, 300, 400, and 500). Finally, each of these sets was compared using a regression model, and the set of features used in the best performing model for each corpus

[5] All corpora use a discrete scale for difficulty level, except for CLEAR, which uses a continuous scale.

[6] We did not consider corpora where perfect performance has been demonstrated (Lee et al., 2021), as this would limit the models' comparison.

[7] Français Langue Maternelle

[8] Français Langue Étrangère

[9] A list of the variables is available at https://cental.uclouvain.be/fabra.

[10] mRMR is a greedy algorithm that chooses the best feature and appends it to the previously selected features on each iteration. The idea is that at each iteration, the algorithm chooses the feature with maximum relevance to classify the target (i.e., univariable classification) and minimum redundancy with the features chosen in previous iterations.

Architecture	WeeBit	OSE	Cambridge
Concatenation $_{\text{BERT, SVM, 54 features}}$ (Imperial, 2021)	-	0.704	-
Concatenation $_{\text{BERT, Log. Regression, 54 features}}$ (Imperial, 2021)	-	0.732	-
Concatenation + Projection $_{\text{BERT, 255 features}}$ (Li et al., 2022)	0.927	0.994	0.877
Soft-Label $_{\text{ROBERTA, Random Forest, 255 features}}$ (Lee et al., 2021)	0.902	0.995	0.752
Soft-Label $_{\text{BART, Random Forest, 255 features}}$ (Lee et al., 2021)	0.905	0.971	0.727
Soft-Label $_{\text{BERT, SVM, 86 features}}$ (Deutsch et al., 2020)	0.877	-	-

Table 1: Summary of F1 measures of readability hybrid models

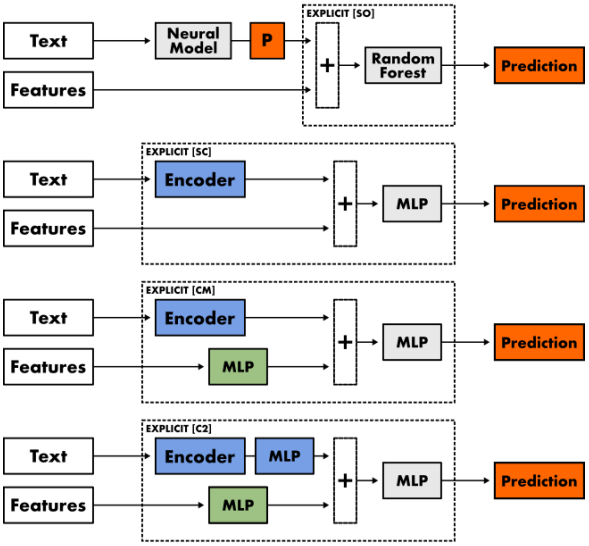
is chosen.[11]

3.3 Models

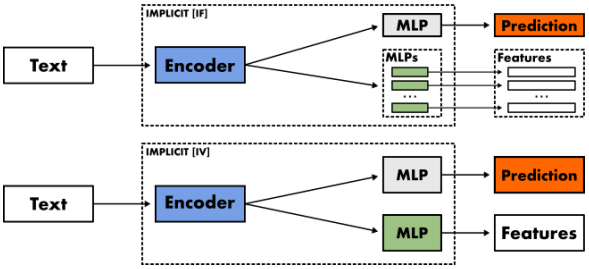
In this work, we explored 8 different architectures[12] (see Figure 1), which may be organized into three groups, based on the features integration method. A key element in the performance of these architectures is the linguistic features to be used. However, considering the different types of corpora explored in this work, it is natural to have different feature sets depending on the language and task. Therefore, the features are considered as a parameter for the architecture.

Baselines (no integration): As a basis for comparison with non-hybrid methods, we considered two baselines that do not combine features with deep learning. The first method, based on deep learning exclusively, uses transformers (henceforth **TR**), more specifically the RoBERTa architecture. This choice was based on the decision to use the same architecture for both languages, where there are fewer models available for French. The second method, based on features exclusively, consists in classical statistical methods. In order to remain consistent with the soft label architecture, we chose to use a **Random Forest (RF)**.

Direct (or explicit) integration: We explored two direct integration methods. The first one is soft-labeling. For the **soft-label (SO)**, we followed the architecture employed by Lee et al. (2021) for readability (see Section 2.2). Note that, in the context of a regression task, there is no difference between soft and hard label. The second method consists in feeding the concatenation between the document encoded by the transformer architecture (i.e. CLS) and the features to the MLP, as in various related



(a) Direct (or explicit) integration (architectures, top to bottom *SO*, *SC*, *CM* and *C2*)



(b) Indirect (or implicit) integration (architectures, top to bottom *IF* and *IV*)

Figure 1: The 6 hybrid architectures explored in this work

works (Section 2). We considered the following flavors of implementation (exemplified in Figure 1a). **Simple Concatenation (SC)**, which simply combines the feature vector with the CLS vector and this concatenated vector feeds the output layer (MLP). In this architecture, the MLP is expected to be able to learn the target along with the mapping between the feature and transformer spaces. By adding an MLP between the features and the concatenation, we could simplify the task by allowing the network to separate the mapping between

[11] We trained the regressor and used its predictions to evaluate the set's quality. In this assessment, we split each corpus into 80% train and 20% evaluation. This split is the same as the first fold of the cross-validation splits used in the models' evaluation.

[12] The range of hyperparameters and the selected values for each architecture are described in Appendix A.

the spaces and/or even create a richer representation of the features. This architecture, here named **Concatenate MLP (CM)**, allows for greater exploration of the search space by adding a few more parameters to the network ($5 \times n$). In the scope of our work, we used an MLP with a first dense layer of $4 \times n$ neurons, followed by a dropout layer (10%), followed by a dense layer of $2 \times n$ neurons that feeds a layer of n neurons (output), where n is the number of features. Following the same idea of including MLPs, the latest variant of the concatenation architecture, **Concatenate 2xMLP (C2)**, also adds an MLP between the encoder output and the concatenation. Thus, the concatenation is performed on the output of two MLPs.

Indirect (or implicit) integration: Language features can also be imprinted on the network through the use of auxiliary tasks, following a multi-task approach. Here, we tested this idea by exploiting the same features used by the concatenation and RF architectures. Alternatively, we could exploit classic NLP tasks as proposed by Zhou et al. (2019), but this would prevent us from controlling indirect features learned by other tasks, making the comparison between the architectures unfair, as this architecture would have access to different information. The first implicit architecture explored in this work, **Implicit Features (IF)**, learns each feature with an independent regression task using an MLP. Thus, the network has $n + c$ output layers (where c is the number of output neurons of the target task; in a regression $c = 1$). Since n can vary depending on the corpus and can have a value considerably higher than c, the network could easily overlook the target task. In order to avoid this possible issue, we considered a weight of 0.5 for the loss associated with the target task and 0.5 for the sum of the other losses. *IF* assumes independence between features, which is not always required. We therefore proposed a simple variation of this architecture to exploit this aspect. In this variant, named **Implicit Feature Vector (IV)**, all the features are grouped into a single output vector of size n. The two implicit models used the same hyperparameter range as the baseline transformers. See Figure 1b for *IF* and *IV* architectures.

4 Results

4.1 Feature Selection

Among the 10 features sets obtained with mRMR, we selected the top features for each corpus based on MSE on the development set. We tested regression with MLP and XGBoost by looking at R^2 and RMSE. As the R^2 of the MLP model was very low in all corpora, we discarded it. See Appendix B for the performance of these models for each set of features. As expected, the feature sets are different for each corpus. Therefore, we use 20, 200, 200 and 500 features respectively for Cambridge, FLE, FLM and CLEAR. The distribution of each feature set with the regression target is showed in Figure 3. We also noticed that no feature is shared between the 4 corpora and only 8 are shared between 3 corpora, all of them illustrating lexical phenomena. Metrics of lexical diversity, such as the Corrected Type-Token Ratio (CTTR) of all types of content words, and verb frequency are observable in both English corpora and respectively the FLM and FLE corpus. The remaining 6 features, shared by the two French corpora and CLEAR, illustrate orthographic neighborhood and 5 different flavors of words imageability, varying only in the way the feature distribution was aggregated (80 percentile, average, interquartile range, kurtosis, and 3rd quartile).

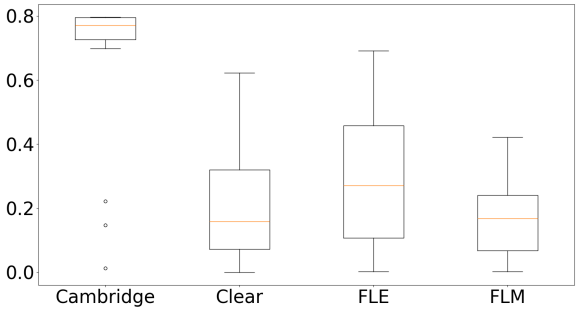


Figure 2: Distribution of the absolute value of correlations between selected features and regression task

4.2 Comparing the performance of the 8 architectures

The results (mean and standard deviation) of the 8 architectures trained in a regression task can be seen in Table 2. The first surprising result is the extremely low R^2 value for some models in the FLM corpus (e.g., *C2* and *IV* for FLM), which means that the model is worse than the average of the regression target. Looking at all results directly, we notice that the *Soft Label* (SO) and *Simple Concatenation* (SC) architectures often have the best results. Looking at the statistical significance, the first conclusion is that the differences in architecture do not generate strong differences between the results.

Table 2: Results by model and corpus. Metrics are average RMSE, MAE, R^2 and R (and standard deviation).

model	RMSE	MAE	R^2	R
		Cambridge		
RF	*0,621 (0,05)*	**0,463 (0,03)**	*0,805 (0,03)*	*0,898 (0,02)*
TR	0,86 (0,15)	0,673 (0,13)	0,62 (0,13)	0,889 (0,02)
C2	0,782 (0,08)	0,665 (0,07)	0,688 (0,07)	0,876 (0,02)
CM	*0,65 (0,07)*	*0,478 (0,04)*	*0,786 (0,04)*	*0,908 (0,02)*
SC	**0,599 (0,09)**	*0,468 (0,07)*	**0,816 (0,05)**	**0,922 (0,02)**
IF	*0,674 (0,16)*	*0,532 (0,17)*	*0,759 (0,13)*	*0,919 (0,01)*
IV	*0,607 (0,1)*	*0,457 (0,07)*	*0,811 (0,06)*	*0,92 (0,02)*
SO	*0,623 (0,05)*	*0,465 (0,03)*	*0,803 (0,03)*	*0,898 (0,02)*
		Clear		
RF	0,669 (0,01)	0,532 (0,01)	0,578 (0,02)	0,764 (0,02)
TR	0,651 (0,05)	0,524 (0,04)	0,598 (0,06)	*0,841 (0,02)*
C2	0,602 (0,05)	0,486 (0,04)	0,657 (0,06)	*0,856 (0,01)*
CM	0,618 (0,03)	0,502 (0,03)	0,64 (0,02)	*0,855 (0,02)*
SC	0,596 (0,02)	0,48 (0,01)	0,665 (0,02)	**0,858 (0,01)**
IF	*0,66 (0,09)*	*0,531 (0,08)*	*0,583 (0,11)*	*0,85 (0,02)*
IV	0,65 (0,08)	0,526 (0,07)	0,595 (0,1)	*0,854 (0,01)*
SO	**0,543 (0,02)**	**0,436 (0,01)**	**0,722 (0,02)**	*0,851 (0,01)*
		FLE		
RF	0,805 (0,02)	0,597 (0,01)	0,681 (0,02)	0,828 (0,01)
TR	0,817 (0,04)	*0,603 (0,04)*	0,671 (0,04)	0,83 (0,02)
C2	0,953 (0,04)	0,77 (0,07)	0,552 (0,04)	0,788 (0,02)
CM	0,872 (0,04)	0,654 (0,03)	0,626 (0,03)	0,828 (0,01)
SC	0,833 (0,08)	*0,62 (0,06)*	0,655 (0,07)	*0,837 (0,01)*
IF	1,089 (0,24)	0,884 (0,24)	0,389 (0,27)	0,7 (0,13)
IV	0,836 (0,04)	0,627 (0,03)	0,656 (0,03)	0,825 (0,02)
SO	**0,749 (0,02)**	**0,572 (0,01)**	**0,724 (0,01)**	**0,855 (0,01)**

model	RMSE	MAE	R^2	R
		FLM		
RF	*2,081 (0,1)*	1,765 (0,07)	*0,358 (0,05)*	*0,652 (0,08)*
TR	2,466 (0,35)	1,964 (0,31)	0,083 (0,25)	0,494 (0,3)
C2	2,638 (0,37)	2,207 (0,33)	*0,000 (0,27)*	*0,691 (0,02)*
CM	*1,995 (0,28)*	*1,625 (0,27)*	*0,400 (0,17)*	**0,726 (0,04)**
SC	*1,996 (0,25)*	**1,575 (0,15)**	*0,402 (0,15)*	*0,697 (0,06)*
IF	*2,333 (0,29)*	1,964 (0,25)	*0,182 (0,20)*	0,615 (0,05)
IV	2,711 (0,24)	2,334 (0,17)	0,000 (0,21)	0,333 (0,35)
SO	**1,988 (0,12)**	*1,687 (0,1)*	**0,414 (0,07)**	*0,692 (0,07)*
		Clear$_{small}$		
RF	**0,728 (0,01)**	*0,58 (0,01)*	*0,501 (0,02)*	0,718 (0,02)
TR	*0,88 (0,14)*	*0,709 (0,12)*	*0,256 (0,22)*	0,622 (0,15)
C2	*0,723 (0,11)*	*0,579 (0,1)*	*0,498 (0,15)*	**0,79 (0,02)**
CM	*0,744 (0,09)*	*0,597 (0,08)*	*0,473 (0,12)*	*0,771 (0,02)*
SC	*0,785 (0,1)*	*0,626 (0,09)*	*0,41 (0,15)*	*0,726 (0,07)*
IF	*0,853 (0,15)*	*0,683 (0,12)*	*0,299 (0,22)*	*0,723 (0,05)*
IV	*0,921 (0,11)*	*0,746 (0,09)*	*0,191 (0,17)*	0,677 (0,05)
SO	**0,728 (0,01)**	**0,579 (0,01)**	**0,501 (0,02)**	0,717 (0,01)
		FLE$_{small}$		
RF	0,887 (0,02)	*0,711 (0,02)*	0,613 (0,02)	0,788 (0,01)
TR	0,944 (0,04)	*0,724 (0,04)*	0,561 (0,04)	0,777 (0,01)
C2	1,278 (0,12)	1,109 (0,09)	0,191 (0,15)	0,731 (0,03)
CM	0,93 (0,04)	0,718 (0,04)	0,574 (0,04)	0,78 (0,02)
SC	0,946 (0,06)	*0,722 (0,06)*	0,559 (0,06)	0,791 (0,01)
IF	1,738 (0,42)	1,407 (0,35)	0,000 (0,60)	0,26 (0,28)
IV	1,1 (0,09)	0,933 (0,1)	0,402 (0,09)	0,718 (0,04)
SO	**0,831 (0,03)**	**0,672 (0,02)**	**0,66 (0,02)**	**0,823 (0,01)**

For example, the only statistically different models for the Cambridge corpus – for all four measures – are *TR* and *C2*. Similarly, for FLM, the *TR* model is the only one with varying performance in the 4 measures, and the R^2 measure has no discriminating power in the statistical analysis of performance of this corpus; furthermore, we observed no difference between *C2*, *SC* and *SO* for the 4 measures. A more distinct trend can be seen with CLEAR and FLE, where *SO* has the best performance (or no statistical difference from the best score) in both corpora for the 4 measures, and, similarly, *IF* for the CLEAR corpus and, on a smaller scale, *SC* for the FLE corpus (where no difference was observed regarding the MAE and R metrics).

Looking at the results in a nutshell, we compared how many times an architecture obtained the best score (or is not statistically different from the best). By combining this information and the evaluation measure, we can calculate how many times on average an architecture was the best. In addition, this measure allows us to group the averages (through the mean) to obtain a single value per architecture. In this way, we found the following values for each architecture: *SO* 3.8, *SC* 2.5, *IF* 2.3, *CM* 2.0 *RF* 1.8, *IV* 1.5, *C2* 0.8, and *TR* 0.5.

Although these values indicate a general ranking, they do not account for the degree of variability in predictions (in other words, a model with a different rank may or may not produce very different predictions). Aiming to shed light on this, we compared the mean of the absolute difference between the scores of the evaluation metrics for all architectures (corpora and models). The top three architectures obtained the following values of RMSE, MAE, R^2 and R respectively: 0.01, 0.05, 0.00 and 0.00 for *SO*, 0.04, 0.17, 0.03 and 0.03 for *SC*, and 0.22, 0.15, 0.19 and 0.19 for *IF*.

One aspect that needs to be studied for a thorough analysis of the results is the impact of corpus size. Indeed, the different corpora we used vary in their number of samples (from 330 to 4,716 samples). To account for this difference, we created subsamples of the 2 largest corpora (respecting the distribution of level and gender), to reach the same number of samples as the two other corpora. We named these subsamples as $\text{FLE}_{\text{small}}$ and $\text{CLEAR}_{\text{small}}$.[13] On these subsamples, we observed that *SO* is the best model in $\text{FLE}_{\text{small}}$, but has no difference from *SC*, *TR* and *RF* (it kept the same tendency except for *RF*). As for $\text{CLEAR}_{\text{small}}$, we observed a remarkable difference where R^2 scores of *IV* and *SO* are now different from the best score, and we can no longer observe significant differences with the other three scores.

Concerning the average ranking of how many times an architecture obtained the best score (or is not statistically different from the best), we note a difference in ranking order, now becoming *SO* 3.5, *SC* 3.0, *CM* and *RF* 2.8, *IF* 2.0, *C2* 1.5, and *TR* 1.0. Despite those differences, the top two are the same.

Studying the absolute mean difference between the evaluation metrics for all architectures, the top three architectures obtained the following values of RMSE, MAE, R^2 and R respectively 0.01, 0.24, 0.00 and 0.00 for *SO*, 0.05, 0.32, 0.05 and 0.05 for *SC*, 0.04, 0.32, 0.04 and 0.04 for *CM*, and 0.04 0.30 0.03 and 0.03 for *RF*. In this scenario, where all the corpora have a small size, there is an improvement in the RF architecture and a considerable reduction in the TR architecture performance (known for its data hunger), where it obtained an average absolute difference of 0.25 for RMSE, 0.82 for MAE, 0.22 for R^2 and 0.22 for R.

This quantitative evaluation allows us to state that the *SO* architecture has the best overall performance, followed by the *SC* architecture, considering the 8 architectures tested. To the best of our knowledge, there are no other studies in the literature that compare those two architectures. Moreover, the existing work on readability is heavily biased towards using classification algorithms, which limits comparison with our results. However, the regression approach applied here allowed us to make proper use of the CLEAR corpus and to account for the ordinal nature of ARA task. In the end, despite the differences, our results are in line with the initial observations in the literature summarized in Table 1.

In summary, we observe that:

- explicit feature integration models outperform implicit ones and baselines;
- the explicit architecture Soft-label (SO) show higher overall performance and the second-best architecture being Simple Concatenation (SC) on both corpora sizes studied;
- the impact of the differences between the architectures is reduce with small corpora, but

[13] The small samples were generated taking into account the distribution of the regression target and the genres.

the ranking of the two best architectures remained the same; and

- statistical machine learning models perform better than the transformers architecture with small corpora.

5 Error analysis

Readability assessment can be strongly influenced by the genre of the documents (Nelson et al., 2012; Dell'Orletta et al., 2014). To investigate this effect in the context of our experiments, we computed the best models' performance scores on each genre of the FLE Corpus (i.e., informative, narrative, dialogue, mail/e-mail and miscellany) and the CLEAR Corpus (i.e., informational and literature). Results are presented in Table 3. We did not observe a clearly stronger impact of genre on one architecture over the other ones, but we have observed that they perform differently for each genre. We noted that models perform consistently well on the informative genre, with an R of approximately 0.85. They perform worst on the miscellaneous genre in the FLE Corpus (R of 0.75 for *SO* and 0.77 for *SC*), which, despite being the biggest sample with 611 texts, is mostly composed of unusual text formats for readability tasks (e.g., poems, menus, songs, and advertisements). On the other end of the scale, the dialogue and mail/e-mail genres (composed of shorter sentences and numerous personal pronouns) show the highest performance scores, especially for the *SO* model. As for the narrative genre, comparable to the latter two in terms of sample size, it is interesting to note that even though the R and R^2 scores are comparable, their RMSE and MAE scores on this genre reveal a statistically poorer performance. This indicates that the order of the levels was learned, but the range was not properly learned.

We also investigated the effect of the task on model performance to assess whether readability predictions could be influenced by the audience (i.e., L1 vs. L2). To ensure a fair comparison between our corpora of different sizes, we used the $\text{FLE}_{\text{small}}$ and $\text{CLEAR}_{\text{small}}$ corpora in this study. Models' performance scores are statistically higher for L2 than for L1 reading (Table 2), which could be explained by several L2 features available in FABRA. Similarly, we compared the performance metrics obtained on English and French corpora and observed that, for the same task (L1 or L2), models perform consistently better on English corpora. The differences observed are striking for the error-based metrics (RMSE and MAE), even though the ranking of architectures remains unaffected for both languages.

Given the large number of features available after the automatic annotation, we investigated the occurrence of features associated with the prediction error of the models. In this study, the feature selection method described in Section 3.2 was used to select the top 100 features associated with error (i.e, statistical residuals). First of all, it is interesting to note that some features used by the models are still correlated with error, hinting that architectures might not have exploited all the information available in the features. The FLM corpus is the most impacted since the intersection between error-related (100 features) and available in the training (200 features) includes 9 features for *SC* and 20 for *SO*. Moreover, we can note that, while lexical features account for roughly half this intersection for both models, discourse features accounts for 30% in *SC*, but for only 17% in *SO*. For each architecture, we then looked at the intersections of these feature lists (error-related and feature set) for the two languages (English and French) and the two tasks (L1 and L2). For the *SC* architecture, the size of the feature intersections for French (10) and English (11) is larger than for L1 (4) and L2 (3). If we compare the two architectures, we observe that the intersections tend to be smaller for *SO* than for the *SC*, suggesting that this model might be able to make better use of the features, which could then be an explanation for his marginal superiority. We also noted that the large proportion of lexical features for French (80% vs. 10% for English) is specific to the *SC* architecture. However, in both models, the intersection for French only includes lexical and syntactic features, and does not include any features related to relationships beyond the sentence level, contrary to English.

6 Conclusion

In this paper, seeking to combine the accuracy of DL with the theory-grounded interpretability of features, we carried out a systematic investigation of how to combine transformers and linguistic features. To this end, we compared 8 different architectures (6 hybrid and 2 baselines) on 4 corpora (in different languages and readability tasks). We observed that a Soft Label architecture obtained the best overall performance, followed by Simple Con-

	INFORMATIVE				NARRATIVE			
Models	RMSE	MAE	R^2	R	RMSE	MAE	R^2	R
SC	0.80 (.13)	0.61 (.10)	0.68 (.11)	0.85 (.04)	0.87 (.21)	0.65 (.14)	0.61 (.20)	0.81 (.08)
SO	0.78 (.04)	0.63 (.03)	0.69 (.03)	0.84 (.02)	0.83 (.11)	0.67 (.10)	0.66 (.09)	0.82 (.06)
	MAIL/EMAIL				MISCELLANY			
Models	RMSE	MAE	R^2	R	RMSE	MAE	R^2	R
SC	0.76 (.07)	0.57 (.05)	0.67 (.07)	0.84 (.02)	0.90 (.06)	0.67 (.04)	0.52 (.07)	0.77 (.02)
SO	0.66 (.05)	0.48 (.04)	0.75 (.04)	0.88 (.03)	0.86 (.03)	0.66 (.03)	0.56 (.03)	0.75 (.02)
	DIALOGUE							
Models	RMSE	MAE	R^2	R				
SC	0.58 (.08)	0.40 (.06)	0.62 (.1)	0.82 (.05)				
SO	0.48 (.05)	0.33 (.05)	0.75 (.06)	0.87 (.03)				

(a) FLE corpus

	INFORMATIVE				LITTERATURE			
Models	RMSE	MAE	R^2	R	RMSE	MAE	R^2	R
SC	0.62 (.04)	0.49 (.03)	0.66 (.04)	0.85 (.02)	0.67 (.06)	0.55 (.05)	0.46(.10)	0.81(.02)
SO	0.56 (.02)	0.45 (.02)	0.72 (.03)	0.85 (.02)	0.53 (.01)	0.42 (.01)	0.66 (.02)	0.82 (.01)

(b) CLEAR corpus

Table 3: Results by genre

catenation. In addition, we explored how language, readability tasks and corpus size impact the performance of these architectures, as well as studying flaws in the use of features by the architectures. The identification of Soft Label as the best architecture is a satisfying result, given that this method is a simple combination of the two proposed baselines, for which several implementations are available. In addition, this result points to an interest for further research into semi-supervised learning in ARA. In addition, our results show several factors associated with the performance of the architectures. Firstly, the size of the corpus can impair the analysis of the difference in performance between the architectures. Second, different types of concatenation may produce better results in specific cases, but overall they perform similarly (overall, Simple Concatenation proved to be the best type of concatenation). Thirdly, implicit architectures have shown some interesting specific results. Given the complexity of these, we suggest that further studies should be carried out in order to explore those approaches. Fourth, traditional ML algorithms, such as RF, are still relevant on small corpora. Finally, transformers, despite being able to maintain some competitive results, are not a silver bullet. As future work, we advocate for further semi-supervised learning studies in ARA and the systematic comparison of hybrid architectures in fields other than ARA.

Limitations

Despite the results pointing to a straightforward solution, they should be taken with a pinch of salt. Firstly, the work focused on a comparison of the architectures, so all the results are based solely on the regression task (differences might be observed in the classification task) and on the same transformer model. Secondly, we searched for the optimal feature set for each corpus from a large set of features. Although realistic, this creates a positive scenario for the contribution of features. Scenarios where the number of features is reduced may lead to different results (e.g. lower performance of hybrid models). In addition, our results are based on four corpora, but each corpus has its own specificities. Although we believe that using more varied corpora than previous similar research is an asset in arriving at robust general conclusions, it is not impossible that, for the discussion on the effect of task and language in Section 5, other corpora would lead to divergent findings. Finally, since our study focuses on ARA, the results may not hold in different fields.

Acknowledgements

Part of this research is supported by the European Commission (Project: iRead4Skills, Grant number: 1010094837. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the

European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This research has been funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under the grant MIS/PGY F.4518.21 and T.0080.23, and also by a research convention with France Éducation International. Par of this research was funded by a FEDtWIN grant (Prf-2020-026-KBR-DLL) funded by BELSPO (Belgian Science Policy). Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

References

Reinald Kim Amplayo, Kang Min Yoo, and Sang-Woo Lee. 2022. Attribute injection for pretrained language models: A new benchmark and an efficient method. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1051–1064.

Ion Madrazo Azpiazu and Maria Soledad Pera. 2019. Multiattentive recurrent neural network architecture for multilingual readability assessment. *Transactions of the Association for Computational Linguistics*, 7:421–436.

Aparna Balagopalan and Jekaterina Novikova. 2020. Augmenting bert carefully with underrepresented linguistic features. *arXiv preprint arXiv:2011.06153*.

M. Cha, Y. Gwon, and H.T. Kung. 2017. Language modeling by clustering with word embeddings for text readability assessment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2003–2006. ACM.

J.S. Chall and E. Dale. 1995. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books, Cambridge.

Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *International Journal of Applied Linguistics*, 165(2):97–135.

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Press Syndicate of the University of Cambridge.

Scott A Crossley, Aron Heintz, Joon Choi, Jordan Batchelor, Mehrnoush Karimi, and Agnes Malatinszky. 2021. The commonlit ease of readability (clear) corpus. In *EDM*.

Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi. 2014. Assessing document and sentence readability in less resourced languages and across textual genres. *ITL-International Journal of Applied Linguistics*, 165(2):163–193.

Tovly Deutsch, Masoud Jasbi, and Stuart Shieber. 2020. Linguistic features for readability assessment. *arXiv preprint arXiv:2006.00377*.

C. Ding and H. Peng. 2003. Minimum redundancy feature selection from microarray gene expression data. In *Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003*, pages 523–528.

Anna Filighera, Tim Steuer, and Christoph Rensing. 2019. Automatic text difficulty estimation using embeddings and neural networks. In *European Conference on Technology Enhanced Learning*, pages 335–348. Springer.

T. François. 2015. When readability meets computational linguistics: a new paradigm in readability. *Revue française de linguistique appliquée*, 20(2):79–97.

Thomas François and Cédrick Fairon. 2012. An "AI readability" formula for French as a foreign language. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 466–477, Jeju Island, Korea. Association for Computational Linguistics.

Daniela B Friedman and Laurie Hoffman-Goetz. 2006. A systematic review of readability and comprehension instruments used for print and web-based cancer information. *Health Education & Behavior*, 33(3):352–373.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Joseph Marvin Imperial. 2021. Knowledge-rich bert embeddings for readability assessment. *arXiv preprint arXiv:2106.07935*.

Ludivine Javourey-Drevet, Stéphane Dufau, Thomas François, Núria Gala, Jacques Ginestié, and Johannes C Ziegler. 2022. Simplification of literary and scientific texts to improve reading fluency and comprehension in beginning readers of french. *Applied Psycholinguistics*, 43(2):485–512.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. 2020. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. *arXiv preprint arXiv:2002.00737*.

J. Kimble. 1992. Plain english: A charter for clear writing. *TM Cooley L. Rev.*, 9:1.

Anna Koufakou, Endang Wahyu Pamungkas, Valerio Basile, Viviana Patti, et al. 2020. Hurtbert: Incorporating lexical features with bert for the detection of abusive language. In *Proceedings of the fourth workshop on online abuse and harms*, pages 34–43. Association for Computational Linguistics.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenbiao Li, Ziyang Wang, and Yunfang Wu. 2022. A unified neural network model for readability assessment with feature projection and length-balanced loss. *arXiv preprint arXiv:2210.10305*.

Fengkai Liu and John SY Lee. 2023. Hybrid models for sentence readability assessment. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 448–454.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Computational Linguistics*, 47(1):141–179.

Farah Nadeem and Mari Ostendorf. 2018. Estimating linguistic complexity for science texts. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 45–55.

J Nelson, C Perfetti, D Liben, and M Liben. 2012. Measures of text difficulty: Testing their predictive value for grade levels and student performance. *Student Achievement Partners*.

Jenny A Ortiz-Zambrano, César Espin-Riofrio, and Arturo Montejo-Ráez. 2022. Combining transformer embeddings with linguistic features for complex word identification. *Electronics*, 12(1):120.

Nicole Peinelt, Marek Rei, and Liakata Maria. 2021. Gibert: Enhancing bert with linguistic information using a lightweight gated injection method. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Maria Soledad Pera and Yiu-Kai Ng. 2014. Automating readers' advisory to make book recommendations for k-12 readers. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 9–16.

Shreya Prabhu, Kara Akhila, and S Sanriya. 2022. A hybrid approach towards automated essay evaluation based on bert and feature engineering. In *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, pages 1–4. IEEE.

Qi Qin, Wenpeng Hu, and Bing Liu. 2020. Feature projection for improved text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8161–8171.

Rudolf Rosa and David Mareček. 2019. Inducing syntactic trees from bert representations. *arXiv preprint arXiv:1906.11511*.

Antony Sare, Aesha Patel, Pankti Kothari, Abhishek Kumar, Nitin Patel, and Pratik A Shukla. 2020. Readability assessment of internet-based patient education materials related to treatment options for benign prostatic hyperplasia. *Academic Radiology*, 27(11):1549–1554.

Andreas Schleicher. 2019. Pisa 2018: Insights and interpretations. *OECD Publishing*.

Andreas Schleicher. 2022. How the european schools compare internationally pisa for schools 2022. *OECD Publishing*.

Veronica Juliana Schmalz and Alessio Brutti. 2021. Automatic assessment of english cefr levels using bert embeddings. In *Proceedings of the Eighth Italian Conference on Computational Linguistics*.

Sowmya Vajjala. 2021. Trends, limitations and open challenges in automatic readability assessment research. *arXiv preprint arXiv:2105.00973*.

Zarah Weiss and Detmar Meurers. 2022. Assessing sentence readability for german language learners with broad linguistic modeling or readability formulas: When do linguistic insights make a difference? In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 141–153.

Rodrigo Wilkens, David Alfter, Xiaoou Wang, Alice Pintard, Anaïs Tack, Kevin P. Yancey, and Thomas François. 2022a. FABRA: French aggregator-based readability assessment toolkit. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1217–1233, Marseille, France. European Language Resources Association.

Rodrigo Wilkens, Daiane Seibert, Xiaoou Wang, and Thomas François. 2022b. MWE for essay scoring English as a foreign language. In *Proceedings of the 2nd Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI) within the 13th Language Resources and Evaluation Conference*, pages 62–69, Marseille, France. European Language Resources Association.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, San Diego, CA. Association for Computational Linguistics.

Kevin Yancey, Alice Pintard, and Thomas Francois. 2021. Investigating readability of french as a foreign language with deep learning and cognitive and pedagogical features. *Lingue e Linguaggio*, 2021(2):229–258.

You Zhang, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2021. Ma-bert: learning representation by incorporating multi-attribute knowledge in transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2338–2343.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.

Zhenyu Zhao, Radhika Anand, and Mallory Wang. 2019. Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. 2019. Limit-bert: Linguistic informed multi-task bert. *arXiv preprint arXiv:1910.14296*.

A Hyperparameters

In this work, we explored two groups of hyperparameters: (1) random forest hyperparameters and (2) transformer hyperparameters. The hyperparameters explored for the soft-label architecture are a combination of the two groups of hyperparameters, while the other hybrid architectures explore the same hyperparameters as transformers. The following hyperparameters were explored:

- Group 1
  - n_estimators: 600, 700, 800 and 900;
  - max_depth: 20, 60 ,100 and None;
  - max_features: sqrt, log2 and None.
- Group 2
  - Learning rate: 1e-2, 1e-3, 1e-4, 1e-5 and 5e-5;
  - Early stop: 1, 3, 5 and 7;
  - Optimizer: adam, sgd;
  - Gradient clipping: no, yes (value of 1)

After exploring the hyperparameters, the following values were chosen for each corpus and architecture:

Corpus	**Architecture**	**n_estimators**	**max_depth**	**max_features**
Clear	RF	600	60	None
	SO	900	None	None
Cambridge	RF	700	None	None
	SO	700	20	None
FLM	RF	800	20	None
	SO	600	100	sqrt
FLE	RF	700	100	sqrt
	SO	800	60	sqrt

Corpus	Architecture	Learning rate	Early stop	Optimizer	Gradient clipping
Clear	TR	0.0001	5	sgd	y
	C2	1e-05	5	adam	y
	CM	1e-05	3	adam	y
	SC	1e-05	1	adam	y
	IF	5e-05	1	adam	y
	IV	1e-05	1	adam	y
	SO	0.0001	5	sgd	y
FLE	TR	5e-05	3	adam	y
	C2	1e-05	1	adam	y
	CM	1e-05	3	adam	y
	SC	5e-05	3	adam	y
	IF	1e-05	3	adam	y
	IV	1e-05	1	adam	y
	SO	5e-05	3	adam	y
FLM	TR	0.0001	1	adam	y
	C2	0.0001	1	adam	y
	CM	0.0001	5	adam	y
	SC	0.0001	3	adam	y
	IF	0.0001	3	adam	y
	IV	0.0001	1	adam	y
	SO	0.0001	1	adam	y
Cambridge	TR	5e-05	1	adam	y
	C2	1e-05	5	adam	y
	CM	5e-05	5	adam	y
	SC	5e-05	5	adam	y
	IF	1e-05	5	adam	y
	IV	1e-05	3	adam	y
	SO	5e-05	1	adam	y

Table 4: Hyperparameters used for each corpus and model

B Details of Feature Selection

Table 5 shows the values of RMSE and R^2 for the number of features. Values in bold are those selected for each corpus. The distribution of the correlations between features and regression target is shown in Figure 3.

#feats	cambridge		clear		FLE		FLM	
	RMSE	R^2	*RMSE*	R^2	*RMSE*	R^2	*RMSE*	R^2
10	0.69	0.78	0.73	0.52	1.02	0.52	1.82	0.48
20	**0.56**	**0.86**	0.73	0.52	0.95	0.59	1.73	0.53
30	0.60	0.83	0.71	0.54	0.92	0.61	1.73	0.53
40	0.67	0.79	0.71	0.55	0.87	0.65	1.55	0.62
50	0.73	0.76	0.72	0.53	0.88	0.64	1.54	0.63
100	0.73	0.76	0.70	0.56	0.87	0.66	1.80	0.49
200	0.69	0.78	0.69	0.56	**0.82**	**0.69**	**1.52**	**0.64**
300	0.65	0.81	0.69	0.57	0.84	0.67	1.75	0.52
400	0.67	0.80	0.69	0.57	0.84	0.68	1.73	0.53
500	0.69	0.78	**0.68**	**0.58**	0.83	**0.69**	1.80	0.49

Table 5: Scores assigned to each set of features for each corpus considering the RSME and R^2 measures

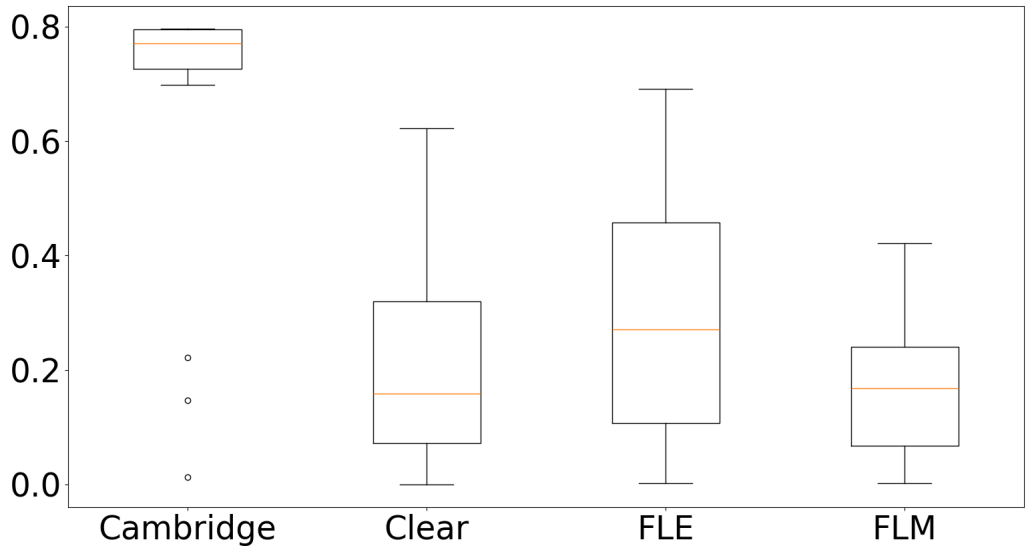


Figure 3: Distribution of correlations between selected features and regression task

Table 6 presents the 8 features from FABRA[14] selected by the model on three corpora.

FEATURE	CLEAR	Cambridge	FLM	FLE
LEXdvrFSC_avg	x	x	x	
LEXfrqCVS_q1	x	x		x
LEXnghFRQH_median		x	x	x
LEXnrmIMG_80P		x	x	x
LEXnrmIMG_avg		x	x	x
LEXnrmIMG_iqr		x	x	x
LEXnrmIMG_kurtosis		x	x	x
LEXnrmIMG_q3		x	x	x

Table 6: Most selected features from FABRA (Wilkens et al., 2022a)

C Models performance by genre

The genres present in each corpora and the number of documents by genre are shown in Table 8.

FLE		**CLEAR**	
Genre	**#**	**Genre**	**#**
Mail/email	135	Literature	2420
Miscellany	611	Informative	2304
Mixed	863		
Dialogue	195		
Informative	414		
Narrative	171		

Table 8: Corpora size separated by gender

[14]https://cental.uclouvain.be/fabra/docs.html