	POS_Features		
1	TTR of word forms	**12**	Numeric_Adj_Tokens
2	Morphemes_word	**13**	Comparative_Adj_Tokens
3	TTR of Lemma	**14**	Conjunction _Tokens
4	Nouns_Tokens	**15**	Conjunction _Subordination _Tokens
5	Verbs_Tokens	**16**	Proper noun_Tokens
6	Adj_Tokens	**17**	Pronoun_Tokens
7	Verb_pseudo_Tokens	**18**	Punc_Tokens
8	Passive verbs_Tokens	**19**	Simple_Connector_Tokens
9	Perfective verbs _Tokens	**20**	Complex_ Connector _Tokens
10	Imperfective verbs _Tokens	**21**	All_Sent_ Connector _Tokens
11	3rdperson_verb_Verbs		
	Syntactic Features		
22	Incidence of subjects	**25**	Incidence of coordination
23	Incidence of objects	**26**	Average phrases/sentence
24	Incidence of modifier/root	**27**	Average phrases depth
	CEFR Word Features		
28	Incidence of Level A1	**32**	Incidence of Level C1
29	Incidence of Level A2	**33**	Incidence of Level C2
30	Incidence of Level B1	**34**	Word entropy with respect to CEFR
31	Incidence of Level B2		
35	**Sentence Embeddings Features**		

Table 3: The Feature set. (all measures are for the rate of tokens on the sentence levels)

lists, 1) Buckwalter and Parkinson 5000 frequency word list based on 30-million-word corpus of academic/non-academic and written/spoken texts (Buckwalter and Parkinson, 2014) KELLY's list which is produced from the Kelly project (Kilgarriff et al., 2014), which directly mapped a frequency word list to the CEFR levels using numerous corpora and languages, 3) lists presented at the beginning of each chapter in 'Al-Kitaab' (Brustad et al., 2015). Merging the lists and aligning them with the Madamira lemmatiser led to our new wide-coverage Arabic frequency list, which can be used to predict difficulty as Entropy of the probability distribution of each label in a sentence. The current list shows some consistency with the English profile list in terms of the percentage of words allocated to each CEFR level.

3.2 Sentence embeddings

In addition to the 34 traditional features we can represent sentences as embedding vectors using different neural models as following:

fastText A straightforward way to create sentence representations is to take a weighted average of word embeddings (WE) of each word, for example, using fastText vectors. This embedding was trained on Common Crawl and Wikipedia using fastText [2] tool. Using the Arabic ar.300.bin file in which each word in WE is represented by the 1D vector mapped of 300 attributes (Grave et al., 2018). We had to normalize the sentence vectors to have the same length with respect to dimensions. For this, we calculated tf-idf weights of each word in the corpus to use them as weights:

$$s = w_1 w_2 \ldots w_n$$

$$Embed.[s] = \frac{1}{n} \sum_i tfidf[w_i] * Embed.[w_i] \quad (1)$$

Universal sentence encoder (Yang et al., 2019) This model requires modeling the meaning of word sequences rather than just individual words. Also it was generated mainly to be used on the sentence level which after sentence tokenization, it encodes sentence to a 512-dimensional vector. We used here the large version[3].

Multilingual BERT (Devlin et al., 2018) Pretrained transformers models proved their ability to learn successful representations of language

[2] https://fasttext.cc/docs/en/crawl-vectors.html

[3] https://tfhub.dev/google/universal-sentence-encoder-multilingual/1

inspired by the transformer model presented in (Vaswani et al., 2017) — who introduced using attention instead to incorporate context information into sequence representation. BERT. Here, we used the last layer produced by BERT transformers while padding the sentences to the maximum length of 128 tokens.

XLM-R (Conneau et al., 2019) This is another multilingual BERT-like model, which is different from mBERT by being trained on Common Crawl (instead of Wikipedias) with slightly different parameters. We used the same setup for classification as in the case of mBERT, while also testing a different setup of combining its output with linguistic features and using it as a joined vector of features for traditional ML classification.

Arabic BERT We trained two available BERT-like pre-trained Arabic transformer models available at Hugging face transformers (AraBERT[4] and Arabic-BERT[5]).

Both models contain both Modern Standard Arabic (MSA) and Dialectal Arabic (DA). The pre-training data used for the AraBERT model consist of 70 million sentences (Antoun et al., 2020). Arabic-BERT trained on both filtered Arabic Common Crawl and a recent dump of Arabic Wikipedia contain approximately 8.2 Billion words(Safaya et al., 2020).

4 Experiments

CEFR language proficiency levels can be presented as labels or as a continuous scale. The former is solved as a classification task with macro-averaged F-1 as the main measure for accuracy. The latter is solved as a regression task (Vajjala and Loo, 2014). At first we decided to work with the three main levels (A,B,and C) because it was quite difficult to determine the boundary between the inner sub levels as in the boundary between B1 and B2.Yet, the other binary classification is either Simple (A+B) or Complex (C). Here there is a problem for evaluation, since the gold standard labels are represented as integers 1, 2, 3 (for the A, B and C levels respectively), which leads to a large number of ties. Out of the standard correlation measures, Kendall's tau-b is designed to handle ties, so in addition to Pearson's ρ this is our measure for regression (Maurice and Dickinson, 1990).

Classification models	P	R	F-1
Features			
KNeighbors	0.51	0.55	0.52
Naive bayes	0.68	0.65	0.65
Decision Tree	0.75	0.77	0.74
Random Forest	0.59	0.75	0.66
XgBoost	0.74	0.77	0.74
Softmax	0.74	0.77	0.74
SVM, Linear	0.75	0.77	0.74
SVM, rbf kernel	**0.75**	**0.77**	**0.75**
Neural			
FastText	0.57	0.59	0.58
UCS	0.52	0.53	0.52
mBERT	0.53	0.54	0.53
ArabicBERT	**0.78**	**0.80**	**0.80**
AraBERT	0.73	0.73	0.73
XLM-R	0.56	0.70	0.61

Table 4: 3-way classification using weighted macro-averaged precision, recall and F-1, Dataset One Using all features versus neural models.

4.1 Readability as a Classification Problem

Table 4 presents the results of classification using updated version of dataset one after application of the error analysis. Applying different ML approaches with 10-fold cross-validation on the 3-way multi-class classification. The classification results as presented in Table 4 divided into two categories: 1) [Linguistics] adding XLM-R vectors to the original set of linguistics features and train with 1058 features [1024 XLM-R dimensions + 34 linguistics features]; 2) [Neural] represents the sentence only by sentence embeddings with neural models.

On the one hand, using linguistic features along with sentence embedding vectors, SVM with rbf kernel classifier provides the best F-1 with 0.75 on the updated corpus version. The SVM classifier is slightly better than both Xgboost and Softmax in precision and they have roughly the same recall value. On the other hand, comparing sentence embeddings of different kinds such as: XLM-R, mBERT, FasText and UCS along with AraBERT and Arabic-BERT.his indicated that Arabic-BERT is a clear winner with F-1 0.80. Since the architecture for building for all BERT-like models are very similar, we suspect that the more Arabic varied

[4]https://huggingface.co/aubmindlab/bert-base-arabert
[5]https://huggingface.co/asafaya/

corpus (Common Crawl and Wikipedia for Arabic-BERT vs Common Crawl XML-R vs Wikipedia for BERT, AraBert and UCS) used to train the Arabic-BERT model is responsible for its better performance

The confusion matrix in Table 5 shows a clear separation between the lower and higher level of proficiency. The majority of errors are between neighbouring levels and the number of errors decreases when we move away from the true class. The most problematic level was B which has a tendency to be classified as CEFR Level A.

Predicted	A	B	C
A	7485	1021	156
B	4506	1112	0
C	0	0	8627

Table 5: Confusion Matrix of SVM (rbf) on 3-way classification with XLM-R.

4.2 Readability as a Regression Problem

Regression allows us to make ranked predictions along the discrete CEFR levels thus assessing which text is more difficult than the other. The training just as in the previous experiment with applying different ML approaches with 10-fold cross-validation. The results for regression can be rated using mean absolute error (MAE) from the gold standard and the correlation coefficients, Pearson, Spearman and Kendall's tau. The results are listed in Table 6. As with classification, error analysis leads to improved results across all methods. The best MAE rate of 0.34 shows that sentence difficulty prediction is quite close to the gold labels. As mentioned before, our model has a very large number of ties for the gold labels (which can only take three values), so the preferred evaluation measure for regression is Kendall's tau-b. The best models are RF and SVR on the XLM-R features.

4.3 Feature Selection

Interpreting feature importance and effectiveness is a way for a better understanding of the classification ML model's logic. This process provides ranking the features by assigning a score for each feature represents its contribution in the target label prediction. These scores provide insights into data representation and model performance. Working with these features ranking can improve the model efficiency and effectiveness by focusing

Model	Pearson	Spearman	Kendall
Decision Tree 2T	0.82	0.62	0.44
Decision Tree 5T	**0.83**	0.64	0.47
Random Forest	**0.82**	0.70	**0.54**
Xgboost	0.78	0.56	0.37
Linear	0.74	0.67	0.49
MLP	0.81	0.68	0.49
SVR,rbf kernel	0.78	0.69	0.52
SVR, Linear	0.8	**0.71**	**0.54**

Table 6: Regression using all features and XLM-R for sentences

only on the important variables and ignore the irrelevant or noisy features. For this purpose, we applied the Recursive Feature Elimination (RFE) a wrapper method for feature selection approach on the basis of SVM classifier. RFE works with recursively removing some features and testing the remain features to select the best feature set affecting the classifier decisions. The results of using RFE approach testing the SVM classifier as represented in Table 7 showing the best ten features contributes to the prediction model. Sentence embedding using XLM-R appeared at the top of the list conveying that it is the most useful feature for sentence difficulty scoring. Followed by the CEFR word frequency features with four features in different positions (Label A1, Label B2, Label C2, and Entropy). The third most effective features are that of the syntactic-set representing more in-depth into the sentence's syntactic knowledge.

35	**Sentence embedding**
26	**Average phrases/sentence**
31	**Incidence of Level B2**
27	Average phrases depth
28	**Incidence of Level A1**
24	Incidence of modifier/root
23	Incidence of objects
22	Incidence of subjects
32	Incidence of Level C1
34	Words CEFR levels entropy

Table 7: List of ten most effective features using REF approach based on SVM classifier

4.4 Ablation

Going further, we performed feature ablation experiments by excluding certain sets of features. We applied SVM rbf classifier on the full dataset