Automatic Difficulty Classification of Arabic Sentences

Nouran Khallaf, Serge Sharoff
School of Languages, University of Leeds
Leeds, LS2 9JT, United Kingdom
`mlnak,s.sharoff@leeds.ac.uk`

Abstract

In this paper, we present a Modern Standard Arabic (MSA) Sentence difficulty classifier, which predicts the difficulty of sentences for language learners using either the CEFR proficiency levels or the binary classification as simple or complex. We compare the use of sentence embeddings of different kinds (fastText, mBERT , XLM-R and Arabic-BERT), as well as traditional language features such as POS tags, dependency trees, readability scores and frequency lists for language learners. Our best results have been achieved using fined-tuned Arabic-BERT. The accuracy of our 3-way CEFR classification is F-1 of 0.80 and 0.75 for Arabic-Bert and XLM-R classification respectively and 0.71 Spearman correlation for regression. Our binary difficulty classifier reaches F-1 0.94 and F-1 0.98 for sentence-pair semantic similarity classifier.

1 Introduction

In the last century, measuring *text readability (TR)* has been undertaken in education, psychology, and linguistics. There appears to be some agreement that TR is the quality of a given text to be easy to comprehend by its readers in adequate time with reasonable effort (Cavalli-Sforza et al., 2018). Research to date has tended to focus on assigning readability levels to whole text rather than to individual sentences, despite the fact that any text is composed of a number of sentences, which vary in their difficulty (Schumacher et al., 2016). Assigning readability levels for a text is a challenging task and it is even more challenging on the sentence level as much less information is available. Also, the sentence difficulty is influenced by many parameters, such as, genre or topics, as well grammatical structures, which need to be combined in a single classifier. Difficulty assessment at the sentence level is a more challenging task in comparison to the better researched text level task, but

the availability of a readability sentence classifier for Arabic is vital, since this is a prerequisite for research on *automatic text simplification* (ATS), i.e. the process of reducing text-linguistic complexity, while maintaining its meaning (Saggion, 2017).

We focus here on experiments aimed at measuring to what extent a sentence is understandable by a reader, such as a learner of Arabic as a foreign language, and at exploring different methods for readability assessment. The main aim of this paper lies in developing and testing different sentence representation methodologies, which range from using linguistic knowledge via feature-based machine learning to modern neural methods.

In summary, the contributions of this paper are:

1. We compiled a novel dataset for training on the sentence level;
2. We developed a range of linguistic features, including POS, syntax and frequency information;
3. We evaluated a range of different sentence embedding approaches, such as fastText, BERT and XLM-R, and compared them to the linguistic features;
4. We cast the readability assessment as a regression problem as well as a classification problem;
5. Our model is the first sentence difficulty system available for Arabic.

2 Corpora and Tools

2.1 Dataset One: Sentence-level annotation

This dataset was used for Arabic sentence difficulty classification. We started building our own dataset by compiling a corpus from three available source classified for readability on the document level along with a large Arabic corpus obtained by Web crawling.

The first corpus source is the reading section of the **Gloss**[1] Corpus developed by the Defense Language Institute (DLI). It has been treated as a gold standard and used in the most recent studies on document level predictions (Forsyth, 2014; Saddiki et al., 2015; Nassiri et al., 2018a,b). Texts in Gloss have been annotated on a six level scale of the Inter-Agency Language Roundtable (IL), which has been matched to the CEFR levels according to the schema introduced by (Tschirner et al., 2015). Gloss is divided according to the four competence areas (lexical, structural, socio-cultural and discursive) and ten different genres (culture, economy, politics, environment, geography, military, politics, science, security, society, and technology).

The second corpus source is the **ALC** , which consists of Arabic written text produced by learners of Arabic in Saudi Arabia collected by (Alfaifi and Atwell, 2013). Each text file is annotated with a proficiency level of the student. We mapped these student proficiency levels to CEFR levels.

Our third corpus source comes from textbook **"Al-Kitaab fii TaAallum al-Arabiyya"** (Brustad et al., 2015) which was compiled from texts and sentences from parts one and two of the third edition but only texts from part three third edition. This book is widely used to teaching Arabic as a second language. These texts were originally classified according to the American Council on the Teaching of Foreign Languages (ACTFL) guidelines which we mapped to CEFR levels.

As these corpora have been annotated on the document level and not on the sentence level, we assigned each sentence to the document level in which it appears, by using several filtering heuristics, such as sentence length and containment, as well as via re-annotation through machine learning, see the dataset cleaning procedure below.

A counterpart corpus of texts not produced for language learners in mind is provided by I-AR, 75,630 Arabic web pages collected by wide crawling (Sharoff, 2006). A random snapshot of 8627 sentences longer than 15 words was used to extend the limitations of C-level sentences coming from corpora for language learners.

Table 1 shows distribution of the number of used sentences and tokens per each Common European Framework of language proficiency Reference [CEFR] Level. In principle we have data for 5-way (A1, A2, B1, etc), 3-way (A, B or C) and

CEFR	Old		New	
	S	T	S	T
A	8661	187225	9030	195343
B	5532	126805	5083	117825
C	8627	287275	8627	287275
Total	22820	601305	22740	600443

Table 1: (S)sentences and (T)tokens available per each CEFR Level in the two versions of the corpus

binary (A+B vs C) classification tasks, but here in this presentation, we focus on the 3-way and binary (simple vs complex) classification tasks.

Dataset cleaning: In our initial experiments we noticed unreliable sentence-level assignments in the training corpus. Therefore, we decided to improve the quality of the training corpus by an error analysis strategy introduced by Di Bari et al. (2014), which is based on detecting agreement between classifiers belonging to different Machine Learning paradigms. The cases when the majority of the classifiers agreed on predicting a label while the gold standard was different were inspected manually by a specialist in teaching Arabic. In our Dataset cleaning experiment we used the following classifiers: SVM (with the rbf kernel), Random Forest, KNeighbors, Softmax and XgBoost using linguistic features discussed in Section 3, trained them via cross-validation and compared their majority vote to the gold standard.

We modified the error classification tags introduced by Di Bari et al. (2014) as follows:

Wrong if the classifiers have wrongly labelled the data, and the gold standard is correct.
Modify if the classifiers are correct and we need to modify the gold standard.
Ambiguous if we consider both either label is possible based on different perspectives.
False is an added label which represent the disagreement between the gold standard and the classifiers, when neither is correct.

For each sentence, five different predictions are assigned. Compared to the gold standard CEFR-label, the classifiers agreed in predicting 10204 instances. Then what we need to consider is when all classifiers agree on the predicted label and it contradicts with the gold standard's one. In that matter, the classifiers agreed on 1943 sentence classification. We manually investigated random sentences and assigned the error classification tags. We found

---
[1]https://gloss.dliflc.edu/

that the main classification confusion was in Level B instances. The analysis results as in Table 4 show the distribution of categories where each error type occurred. In the end, 380 instances had to be assigned to lower level (usually from B to A).

2.2 Dataset Two: Simplification examples

A set of simple/complex parallel sentences has been compiled from the internationally acclaimed Arabic novel **"Saaq al-Bambuu"** (Al-Sanousi, 2013) which has an authorized simplified version for students of Arabic as a second language (Familiar, 2016). We assume that a successful classifier should be able to detect sentences in the original text that require simplification. Dataset Two consists of 2980 parallel sentences Table 2.

Level	Sentence	Token
Simple A+B	2980	34447
Complex C	2980	46521
Total	5690	80968

Table 2: Number of Sentences and Tokens available per each CEFR Level in Dataset two

3 Features and extraction methods

We work with following groups of features in Table 3: Part of speech tagging features (POS-features); Syntactic structure features (Syntactic-features); CEFR-level lexical features; Sentence embeddings.

3.1 Linguistic features

While the sentence-level classification task is novel, we borrowed some features from previous studies of text-level readability (Forsyth, 2014; Saddiki et al., 2015; Nassiri et al., 2018a,b). We decided to exclude the sentence length from the feature set, as this creates an artificial skew in understanding what is difficult: more difficult writing styles are often associated with longer sentences, but it is not the sentence length which makes them difficult. Specifically, many long Arabic sentences contain shorter ones, which are connected by conjunctions

such as ' /wa /= and'. According to the experience of language teachers such sentences do not present problems for the learners.

3.1.1 The POS-features

[Table 3 features (1-21)], these features represent the distribution of different word categories in the sentence, and the morpho-syntactic features of these words. According to Knowles and Don (2004), Arabic lemmatization, unlike that of English, is an essential process for analysing Arabic text, because it is a methodology for dictionary construction. Therefore, we used the Lemma/Type ratio instead of Word/Type ratio. Adding features represents the different verb types (Verb pseudo, Passive verbs, Perfective verbs, Imperfective verbs and 3rdperson). As conjunction is one of the important features in representing sentence complexity in Arabic (Forsyth, 2014), we used the annotated discourse connectors introduced by Alsaif (2012) by splitting this list into 23 simple connectors and 56 complex connectors referring to non-discourse connectors and discourse connectors respectively. For POS-features extraction we used MADAMIRA a robust Arabic morphological analyser and part of speech tagger (Pasha et al., 2014).

3.1.2 Syntactic features

Features (22-27) from Table 3 provide some information about the sentences structures and number of phrases as well as phases types. These features are derived from a dependency grammar analysis. Because dependency grammar is based on word-word relations, it assumes that the structure of a sentence consists of lexical items that are attached to each other by binary asymmetrical relations, which is known as dependency relations. These relations will be more representative for this task. We used CamelParser (Shahrour et al., 2016) a system for Arabic syntactic dependency analysis together with contextually disambiguated morphological features which rely on the MADAMIRA morphological analysis for more robust results.

3.1.3 CEFR-level lexical features

Features (28-34) from Table 3 are used to assign each word in the sentence with an appropriate CEFR level. For this, we created a new Arabic word list consisting of 8834 unique lemmas labelled with CEFR levels. This list was a combination of three frequency lists, 1) Buckwalter and Parkinson 5000 frequency word list based on

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)

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) Pre-trained transformers models proved their ability to learn successful representations of language inspired by the transformer model presented in

---
[2]https://fasttext.cc/docs/en/crawl-vectors.html
[3]https://tfhub.dev/google/universal-sentence-encoder-multilingual/1

(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).

---

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

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 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.

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

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 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 while excluding one of the four main group of features blocks POS, Syntactic-features, CEFR-level lexical features, sentence embedding, along with using

only the sentence embeddings (XLM-R since this was shown in comparison of embeddings below). This shows that the sentence embeddings significantly contribute to the classification results Table 8), in spite of the efforts to create hand-crafted features. Nevertheless, the linguistic features are useful in interpreting the results of purely neural classification. This results prove that the transformer models provide a rich representation for the sentences covering linguistic features.

According to the primary results from the feature selection and the ablation experiments, which proved the use of sentence embedding alone could fulfill the task without extensive use of linguistics features. These results encouraged us to continue experimentation with applying only the sentence embedding feature to reduce the number of features which consequently decreases the data analysis and training time.

Feature set	P	R	F-1
Exclude XLM-R	0.49	0.63	0.55
Exclude POS	0.55	0.71	0.62
Exclude Syntactic	0.57	0.69	0.59
Exclude CEFR	0.55	0.71	0.62
Only XLM-R	0.75	0.77	**0.75**

Table 8: SVM Classification ablation experiment on 3-way classification

4.5 Testing on Dataset Two

For the binary classification, the classifier reached F-1 of 0.94 and 0.98 for Arabic-BERT and SVM XML-R respectively. However, when testing the binary classifiers trained from DataSet One on Dataset Two the accuracy drops considerably, see Table 9.

As the confusion matrix in Table 10 shows, both classifiers performed better in identifying the complex instances rather than simple ones, so the F1 measure drops. However, the initial results on dataset two shows that XLM-R classifier performed better than Arabic-BERT, we still consider Arabic-BERT classifiers [both 3-way and binary] as best classifier so far. Our interpretation for these confusions is because of the fictional nature of Dataset Two. First, the fiction is well represented in the training data for the A+B levels in Dataset One, while the C level (Snapshot corpus) contains texts of many different types from the internet, so that the classifiers could not handle the mismatch in

	Arabic-BERT	XLM-R
P	0.60	0.56
R	0.50	0.53
F-1	0.53	0.54

Table 9: Fine-tuned Arabic-BERT versus SVM XLM-R Classifier's performance on Dataset two

	ArabicBert		XLM-R	
Predicted	**A**	**C**	**A**	**C**
A	19	2961	138	2842
C	46	2934	223	2757

Table 10: Confusion Matrix with binary classifier Arabic-BERT versus XLM-R on Dataset Two.

genres. The other possible reason is that what is considered as complex sentences which are worth simplification according to developers of Dataset Two does not really seem to be complex as to be only suitable for the C-level students. More research is needed to identify the difference between the two datasets.

4.6 Dataset Two Sentence Similarity

For the purpose of this experiment we needed to include non-simplified sentence to the Dataset two. So that, we duplicated the 2980 complex sentences without simplification aligned with the exact sentence without modification and labeled them with 0 indicating not paraphrased/simplified. Resulting in a Dataset consist of 2980 sentences with right simplification labeled as 1 and other 2980 non-simplified with label 0, in a total of 5960 sentence. The two models trained on this similarity task (AraBert and Arabic-Bert) achieve the F-1 measure of 0.98, leading to the ability to detect sentences which need simplification according to the Dataset Two standard.

5 Related Work on Arabic

The last two decades have seen enormous efforts (especially for the English language) to develop readability measurement ranging from the traditional readability formulae to ML algorithms. English language researchers have introduced more than 200 readability formulae (DuBay, 2004) as well as hundreds of models (Schwarm and Ostendorf, 2005). In contrast, less research has addressed Arabic language issues and their challenges for robust readability formulae.

Some attempts to formulate statistical formulae

for the Arabic language reflected traditional English formulae such as the Flesch–Kincaid Grade. The simplest formulae included the average word length, the average sentence and other surface features. According to Cavalli-Sforza et al. (2018) these simple formulae are Dawood formula (1977), Al-Heeti formula (1984), and the formula presented by Daud et al. (2013) based on a corpus. The more sophisticated formulae represent the syllables and more insights the Arabic sentences, such as AARI Base by Al Tamimi et al. (2014) and OSMAN by El-Haj and Rayson (2016).

Other studies were conducted to measure text readability by targeting either first or second language learners for Arabic language modelled using different ML algorithms. Most of these studies used the previously traditional features along with varying lists of part of speech features (POS) representing the words in each document as in studies by (Al-Khalifa and Al-Ajlan, 2010; Forsyth, 2014; Saddiki et al., 2015; Nassiri et al., 2018a). Forsyth (2014) used the word frequency dictionary by Buckwalter and Parkinson (2014) to classify the words' level against this dictionary frequencies. The dictionary was used later by Nassiri et al. (2018b) along with 133 POS features to achieve an accuracy of 100% with 3-classes. The 'Al-Kitaab' textbook has a word list introduced at the beginning of each chapter in the book. These lists were used by Cavalli-Sforza et al. (2014) for comparing the words appeared in a text against this list and labelling them by (target, known, unknown). Saddiki et al. (2018), highlight adding new syntactic features to their features targeting more in-depth analysis. They used two different datasets for both first and second Arabic language learning. This yielded an accuracy of 94.8%, 72.4% for first language learners and second language learners respectively.

6 Conclusions

We present the first attempt to build a methodology for Arabic difficulty classification on the **sentence** level. We have found that while linguistic features, such as POS tags, syntax or frequency lists are useful for prediction, Deep Learning is the most important contribution to performance, but the traditional features can help in interpreting the black box of Deep Learning alone. For this specific task and for the Arabic language, fine-tuned Arabic-BERT offers better performance than other sentence embedding methods. Also, application of the classifiers trained on one dataset to a very different evaluation corpus shows that the classifiers learn some important properties of what is difficult in Arabic, but the transfer is more successful for the feature-based models than for the BERT-based ones.

In the end, our best classifier is reasonably reliable in detecting complex sentences; however, it is less successful in separating between the lower learner levels. Still the binary classifier provides the functionality for filtering out really difficult sentences, not suitable for the learners. If we are thinking of Arabic learners especially in higher education, we are expecting learners to graduate with a BA degree in the case of Arabic as a complex language with confidence in reading B2 texts, which implies that the tool for separating A+B vs C level texts is really useful for undergraduate teaching.

Through our tool providing computational assessment of difficulty, we will be able: i) to select the appropriate texts for students; ii) to access ever-larger volumes of information to find educational material of the right difficulty online; iii) to explore curriculum-based assessment to find what is most effective in finding gaps in a curriculum that can be filled according to students' needs.

Our future work involves building a parallel simple/complex Arabic corpus for sentence simplification. The corpus will be classified on the basis of how difficult the sentences are in a Common Crawl snapshot of Arabic web pages. Using the text difficulty classifier, we can split the corpus into two groups for complex and simple sentences. We also consider the semantic similarity detection on **"Saaq al-Bambuu"** as a benchmark, which could be used in the corpus compilation. In this study we only performed some ablation analysis, but because BERT-like models are more useful as the classifiers, we want to investigate their performance via probing for linguistic features following the BERTology framework (Rogers et al., 2020; Sharoff, 2021). We also want to explore the link between the difficulty assessment on the document vs sentences levels (Dell'Orletta et al., 2014).

References

Hend S Al-Khalifa and Amani A Al-Ajlan. 2010. Automatic readability measurements of the arabic text: An exploratory study. *Arabian Journal for Science and Engineering*, 35(2 C):103–124.

Saud Al-Sanousi. 2013. *Saaq al-Bambuu*. Arab Scientific Publishers Inc., Lebanon.

Abdel Karim Al Tamimi, Manar Jaradat, Nuha Al-Jarrah, and Sahar Ghanem. 2014. Aari: automatic arabic readability index. *Int. Arab J. Inf. Technol.*, 11(4):370–378.

A. Alfaifi and Eric S Atwell. 2013. Arabic learner corpus v1: A new resource for arabic language research. In *In proceedings of the Second Workshop on Arabic Corpus Linguistics (WACL-2)*. Leeds.

Amal Alsaif. 2012. *Human and automatic annotation of discourse relations for Arabic*. University of Leeds.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

K Brustad, M Al-Batal, and A Al-Tonsi. 2015. *Al-Kitaab fii Tacallum al-cArabiyya. A Textbook for Beginning Arabic: Part One Third Edition*. Georgetown University Press.

Tim Buckwalter and Dilworth Parkinson. 2014. *A frequency dictionary of Arabic: Core vocabulary for learners*. Routledge.

Violetta Cavalli-Sforza, Mariam El Mezouar, and Hind Saddiki. 2014. Matching an arabic text to a learners' curriculum. In *Proc. 5th Int. Conf. on Arabic Language Processing (CITALA), Oujda, Morocco*, pages 79–88.

Violetta Cavalli-Sforza, Hind Saddiki, and Naoual Nassiri. 2018. Arabic readability research: Current state and future directions. *Procedia computer science*, 142:38–49.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Nuraihan Mat Daud, Haslina Hassan, and Normaziah Abdul Aziz. 2013. A corpus-based readability formula for estimate of arabic texts reading difficulty. *World Applied Sciences Journal*, 21:168–173.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Marilena Di Bari, Serge Sharoff, and Martin Thomas. 2014. Multiple views as aid to linguistic annotation error analysis. In *Proceedings of LAW VIII-The 8th Linguistic Annotation Workshop*, pages 82–86.

William H DuBay. 2004. The principles of readability. *Online Submission*.

Mahmoud El-Haj and Paul Rayson. 2016. Osman—a novel arabic readability metric. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 250–255.

Laila Familiar. 2016. *Saud al-Sanousi's Saaq al-Bambuu: The Authorized Abridged Edition for Students of Arabic*. Georgetown University Press.

Jonathan Neil Forsyth. 2014. *Automatic readability prediction for modern standard Arabic*. Ph.D. thesis, Brigham Young University. Department of Linguistics and English Language.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) [Online]*.

Adam Kilgarriff, Frieda Charalabopoulou, Maria Gavrilidou, Janne Bondi Johannessen, Saussan Khalil, Sofie Johansson Kokkinakis, Robert Lew, Serge Sharoff, Ravikiran Vadlapudi, and Elena Volodina. 2014. Corpus-based vocabulary lists for language learners for nine languages. *Language resources and evaluation*, 48(1):121–163.

Gerry Knowles and Zuraidah Mohd Don. 2004. The notion of a "lemma": Headwords, roots and lexical sets. *International Journal of Corpus Linguistics*, 9(1):69–81.

Kendall Maurice and Gibbons Jean Dickinson. 1990. Rank correlation methods. *London: Edward Arnold*.

Naoual Nassiri, Abdelhak Lakhouaja, and Violetta Cavalli-Sforza. 2018a. Arabic readability assessment for foreign language learners. In *International Conference on Applications of Natural Language to Information Systems*, pages 480–488. Springer.

Naoual Nassiri, Abdelhak Lakhouaja, and Violetta Cavalli-Sforza. 2018b. Modern standard arabic readability prediction. In *International Conference on Arabic Language Processing*, pages 120–133. Springer.

Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for

morphological analysis and disambiguation of arabic. In *Lrec*, volume 14, pages 1094–1101.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Hind Saddiki, Karim Bouzoubaa, and Violetta Cavalli-Sforza. 2015. Text readability for arabic as a foreign language. In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE.

Hind Saddiki, Nizar Habash, Violetta Cavalli-Sforza, and Muhamed Al Khalil. 2018. Feature optimization for predicting readability of arabic l1 and l2. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 20–29.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059.

Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 10(1):1–137.

Elliot Schumacher, Maxine Eskenazi, Gwen Frishkoff, and Kevyn Collins-Thompson. 2016. Predicting the relative difficulty of single sentences with and without surrounding context. *arXiv preprint arXiv:1606.08425*.

Sarah Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 523–530, Ann Arbor, Michigan. Association for Computational Linguistics.

Anas Shahrour, Salam Khalifa, Dima Taji, and Nizar Habash. 2016. Camelparser: A system for arabic syntactic analysis and morphological disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 228–232.

Serge Sharoff. 2006. Open-source corpora: using the net to fish for linguistic data. *International Journal of Corpus Linguistics*, 11(4):435–462.

Serge Sharoff. 2021. Genre annotation for the web: text-external and text-internal perspectives. *Register studies*, 3.

E Tschirner, O Bärenfänger, and K Wisniewski. 2015. Assessing evidence of validity of the actfl cefr listening and reading proficiency tests (lpt and rpt) using a standard-setting approach. *Technical Report 2015-EU-PUB-2*.

Sowmya Vajjala and Kaidi Loo. 2014. Automatic cefr level prediction for estonian learner text. In *Proceedings of the third workshop on NLP for computer-assisted language learning*, pages 113–127.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*.