Pixels at BAREC Shared Task 2025: Visual Arabic Readability Assessment

Ben Sapirstein
Reichman University
ben.sapirstein@post.runi.ac.il

Abstract

We present a visual-language approach to Arabic readability assessment using the PIXEL Vision Transformer, which processes rendered text as images to bypass tokenization challenges. Our system participated in the BAREC 2025 Shared Task (Sentence-level Strict track). We evaluate orthographic variants (normalization, diacritization, transliteration) and morphological segmentation with different visual boundary markers. Results show that diacritization provides useful visual cues for disambiguation, morphological segmentation improves over word-level processing, and transliterated scripts outperform native Arabic script. Our approach demonstrates the potential of visual processing for readability assessment in complex languages and writing systems.

1 Introduction

Text readability is fundamental to effective comprehension, retention, reading speed, and engagement, with texts exceeding a reader's ability often leading to disengagement and frustration (DuBay, 2004). For Arabic, a language spoken by over 400 million people worldwide, developing robust readability assessment models is crucial for advancing literacy, language learning, and academic performance (Elmadani et al., 2025b). These models are essential for educators to prepare appropriate reading materials and enhance the learning experience, making complex concepts accessible to a wide range of students across the Arab world's linguistically diverse populations. Arabic readability assessment presents significant challenges rooted in the language's morphological richness, dialectal variants, orthographic ambiguity and inconsistency (Habash, 2010), and the profound implications of these complexities on standard tokenization methods.

We introduce an alternative approach: treating text as a visual signal. By rendering Arabic sentences as images and processing them with the PIXEL Vision Transformer (Rust et al., 2022), we aim to capture readability cues directly from the graphetic and typographical properties of the text. This approach offers several advantages: (1) It bypasses the vocabulary bottleneck of token-based models, avoiding sparsity and tokenization errors; (2) It naturally encodes orthographic and morphological variation; and (3) It facilitates cross-language and cross-script transfer from large-scale pretraining.

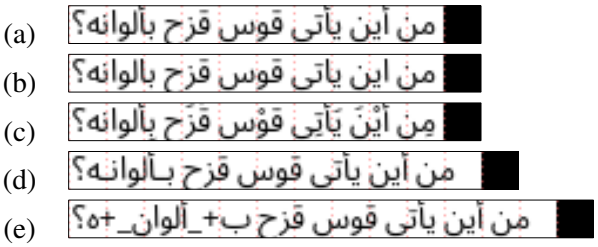

Figure 1: Visual comparison of Arabic script input variants, from top: (a) Default, (b) Normalized, (c) Diacritized, (d–e) Morphological segmentation (Tatweel and default).

We describe our submission to the BAREC 2025 Shared Task on Arabic readability assessment (Elmadani et al., 2025a), where we: (1) Apply PIXEL to sentence-level Arabic readability; (2) Compare orthographic variants including normalization, diacritization, and transliteration; (3) Evaluate morphological segmentation schemes with different visual boundary markers.

Our experiments reaffirm PIXEL's robustness on orthographic variance and reveal that diacritization provides beneficial visual disambiguation cues, morphological segmentation can improve performance, and transliterated scripts yield more tractable visual patterns. The findings highlight the potential of visual processing for readability assessment in complex languages and writing systems.

Proceedings of The Third Arabic Natural Language Processing Conference, pages 350–356

2 Background

2.1 Arabic Readability Assessment

The Arabic readability assessment landscape features several important datasets and frameworks. Taha-Thomure (2017) developed a 19-level text leveling framework for children's literature, adopted by the Arab Thought Foundation's Arabi21 initiative to tag over 9,000 books. This procedural framework outlines ten qualitative and quantitative criteria, including text genre, abstractness of ideas, vocabulary, text authenticity, and sentence structure, primarily targeting full texts and early education

The SAMER project contributed a five-level readability lexicon for Modern Standard Arabic (Al Khalil et al., 2020), initially containing 26,000 lemmas and later expanded to more than 40,000. The lexicon was manually annotated in triplicate by language professionals from three regions of the Arab world and with detailed annotation guidelines. SAMER also produced the first manually annotated Arabic text simplification corpus (Alhafni et al., 2024), 159K words from 15 fiction novels with document- and word-level annotations. These efforts are supported by practical applications such as the Google Docs add-on by Hazim et al. (2022), which visualizes word-level readability to assist human annotators in text simplification

Leveraging the SAMER project resources, Liberato et al. (2024) systematically explored different modeling approaches for Arabic readability assessment, ranging from rule-based methods to Arabic pretrained language models. Their research benchmarked models on a newly created corpus at both word and sentence fragment levels, highlighting the challenges posed by Arabic's morphological richness and limited readability resources. Their findings demonstrated that combining different modeling techniques yielded the best results.

Further extending these initiatives, the Balanced Arabic Readability Evaluation Corpus (BAREC) (Elmadani et al., 2025b) provides a large-scale, fine-grained dataset consisting of 1,922 documents with 69,441 sentences spanning over 1 million words. This corpus is carefully curated to cover 19 readability levels, from kindergarten to postgraduate comprehension, balancing genre diversity, topical coverage, and target audiences. BAREC is considered the largest and most fine-grained manually annotated Arabic readability resource to date (Habash et al., 2025).

2.2 Arabic Processing Challenges

Arabic poses major challenges for NLP tasks.

Morphological richness is a significant characteristic, entailing complex inflections and cliticization. Arabic words inflect for numerous grammatical features such as gender, number, person, case, aspect, mood, and voice, while also incorporating various attachable proclitics (e.g., conjunctions, prepositions, definite article) and enclitics (e.g., pronominal objects) (Liberato et al., 2024). This complexity leads to an extensive number of word forms; for example, Modern Standard Arabic (MSA) verbs alone can have upwards of 5,400 forms (Obeid et al., 2020). Such morphological complexity results in lexical sparsity and significantly complicates tasks like tokenization. In fact, Arabic exhibits a vocabulary growth rate approximately 2.5 times higher and out-of-vocabulary rates about 10 times higher than English (Habash, 2010).

Dialectal variations further complicate Arabic processing. While MSA is the formal written standard used in education, media, and literature across the Arab world, it is not the native language of any Arab speaker. Instead, native speakers communicate using a diverse array of informal spoken dialects that differ considerably from MSA and from each other in their phonology, morphology, lexicon, and even syntax (Habash, 2010). A key issue is the general lack of standardized spelling systems for Arabic dialects, which contributes to orthographic inconsistency. For instance, different forms of the letter Alif (ا، أ، إ، آ) can represent the same linguistic unit: the common writing راس instead of the standard رأس results in different character codes despite conveying the same word. Orthographic normalization addresses this issue by converting letter variants or visually similar letters into a single, standardized form (Obeid et al., 2020). At the same time, informal sociolinguistic norms often guide how dialects are written, and NLP systems must be able to recognize and adapt to these conventions to fully leverage the information such texts provide.

Orthographic ambiguity is a pervasive problem in written Arabic. This means that a single written form can correspond to multiple different meanings and grammatical analyses. For example, the word درسها (drshA) can be interpreted in several ways depending on the implied diacritics: as a verb meaning 'he taught her', another verb meaning 'he studied it', or a noun phrase meaning 'her lesson'. While automatic disambiguation methods, such as Maximum Likelihood Estimation (MLE) disambiguators (Khalifa et al., 2016), attempt to resolve this issue by inserting diacritical marks that specify short vowels and consonantal geminations, the resulting proliferation of unique tokens further intensifies lexical sparsity and adds to the vocabulary bottleneck already posed by Arabic's morphological richness.

Script complexity and allographic variation pose additional challenges for visual processing. The Arabic script provides multiple different graphs that can represent the same letters as in contextual forms (e.g Ayin variants ع ، ع ، ع ، ع), multi-character ligatures and complex word-level ligatures. While Unicode normalization can be applied to avoid inflated token vocabularies (Obeid et al., 2020), standard font features will map even Unicode-standardized input to different graphs, leading to visual variation.

Transliteration schemes such as Buckwalter (BW) and Habash-Soudi-Buckwalter (HSB) (Habash et al., 2007), offer an alternative approach to handling Arabic's orthographic complexity. HSB is particularly beneficial for visual processing, as different Arabic letter variants are mapped to visually similar Latin glyphs while preserving the orthographic distinctions of the source script (Figure 2). Additionally, Latin-based representations present fewer rendering challenges since they do not exceed typical line boundaries, unlike certain Arabic diacritics and punctuation marks.

2.3 Visual Embeddings for Language

The PIXEL model (Rust et al., 2022) treats text as images by rendering text in fixed fonts and processing image patches through Vision Transformers (Dosovitskiy et al., 2020). PIXEL is built upon the architecture of Masked Autoencoders (He et al., 2021), which are scalable self-supervised learners that use an asymmetric encoder-decoder design and masking to reconstruct missing image pixels for efficient visual representation learning. PIXEL

(a) إنه شقيقى الأصغر

(b) إنَّهُ شَقِيقيَّ الأَصْغر

(c) <nh $qyqy Al>Sgr

(d) <in~ahu $aqiyqay~a Al>aSogari

(e) Ănh šqyqy AlÂSγr

(f) Ăin~ahu šaqiyqay~a AlÂaS.γari

Figure 2: Visual comparison of script variants before and after diacritization: (a,b) Arabic script, (c,d) Buckwalter, (e,f) HSB. Transliterated forms properly display all diacritic information, with HSB maintaining intuitive visual representations of Arabic letter variants such as different Alif forms (lA, $\mathit{Â}$, $\mathit{Ă}$) .

has demonstrated strong performance as a foundation model across various languages and scripts, including Arabic, where it achieves near-parity with token-based models on core NLP tasks (95.7% vs. 95.4% POS tagging accuracy compared to BERT; 77.3 vs. 77.7 LAS in dependency parsing).

The PIXEL model addresses some of the mentioned challenges: orthographic variations often appear as visually similar glyphs, and the visual representation allows accessing morphemes without tailored tokenization. This continuous vocabulary representation is particularly useful for dialectal data, as demonstrated by experiments on German dialects (Muñoz-Ortiz et al., 2024). However, there is still a potential pitfall when processing allographs.

This approach naturally handles RTL scripts, though Rust et al. (2022) note processing limitations where RTL sentences are processed from end to beginning, potentially affecting positional learning.

2.4 BAREC Shared Task 2025

The BAREC Shared Task 2025 focuses on fine-grained Arabic readability assessment, participants in the shared task are challenged to build models for both sentence-level and document-level readability classification.

A strong baseline for this task, as established in the research accompanying the BAREC corpus, is AraBERTv2 (Antoun et al., 2020). This model, when used with the D3Tok input variant and Cross-Entropy loss, achieved the best performance across various metrics in initial benchmarking experiments. We compare our results to the Word input variant.

3 System Overview

Our pipeline begins by rendering each Arabic text as an RGB image. We use Noto Sans Arabic at a fixed font size on a white background, following the standard PIXEL methodology (Rust et al., 2022). Sentences are rendered to a fixed image size determined by the patch size and the maximum sequence length. The image is then split into non-overlapping 16×16 patches, each patch is flattened and linearly projected to the ViT encoder. For fine-tuning, we append a linear classification head, with softmax and cross-entropy loss.

4 Experimental Setup

4.1 Text Processing Variants

We introduce two dimensions of preprocessing:

Orthographic encoding manipulates surface forms to test the effect of script and phonological cues. For Arabic script, we evaluate the individual effects of (i) dediacritization and (ii) orthographic normalization, as well as their combination, and compare them to the default and fully diacritized forms (via CAMeL's MLE disambiguator). For transliterated scripts we restrict evaluation to three variants (default, normalized+dediacritized, and diacritized).

Morphological encoding manipulates word structure. Using CAMeL Tools' MLE-based tokenizer, we segment words into stems and clitics (e.g., وكتابها → و + كتاب + ها). To make these boundaries visually salient, we experiment with different markers: standard ASCII markers (+_ and _+), Arabic tatweel to maintain script consistency, and spaces treating morphemes as distinct visual units (Figure 1).

4.2 Evaluation Metrics

We report results on Accuracy, ±1 Accuracy, MAE, and Quadratic Weighted Kappa (QWK) as the primary metric which measures agreement while accounting for the ordinal distance between predicted and true levels.

5 Results and Analysis

5.1 Orthographic Encoding Effects

Table 1 summarizes the impact of orthographic variants across Arabic, Buckwalter, and HSB scripts. A consistent pattern emerges: transliterated scripts outperform Arabic script across all metrics, with HSB achieving the highest QWK (69.3%), followed by Buckwalter (68.0%), while Arabic peaks at 66.5%. This "script gap" of approximately 3-4 QWK points suggests that visual regularity in Latin-based representations provides advantages for the vision transformer architecture.

Within each script, preserving orthographic and diacritic distinctions generally benefits PIXEL performance more than normalization. Diacritization shows particular promise for transliterated scripts, improving QWK by 1.3 points for Buckwalter and 2.4 points for HSB. However, diacritization effects in Arabic script are mixed, possibly due to incomplete visual rendering of diacritical marks that extend beyond typical line boundaries.

5.2 Morphological Encoding Effects

Table 2 presents the impact of morphological segmentation on readability assessment. Morphological segmentation using D3TOK generally improves performance over word-level processing, with both tatweel and space markers achieving 67.4% and 67.0% QWK respectively, compared to 66.3% for unsegmented text. The standard ASCII markers under-perform the baseline word-level approach. The effectiveness of space separation is particularly noteworthy, despite spaces already serving as word boundaries in the text.

5.3 Official Results

For official submission, we submitted the predictions of the default Arabic script variant. Table 3 shows that our model achieved 68.4% QWK on the blind test. However, PIXEL significantly underperformed the AraBERTv2 baseline, which achieved 76.2% QWK.

6 Conclusion and Future Work

PIXEL naturally handles orthographic variation while benefiting from morphological and phonological signals in richer text representations. English pretraining benefits from Latin script regularity, though the performance gap with token-based models suggests need for further optimization.

Script	Configuration	Accuracy	±1 Acc	MAE	QWK
Arabic	Default	40.0%	53.0%	1.74	**66.5%**
	Dediacritized	41.0%	53.7%	1.74	63.9%
	Ortho Normalized	38.8%	51.5%	1.79	65.6%
	Ortho Normalized & Dediacritized	40.0%	53.3%	1.73	64.8%
	Diacritized	41.7%	54.7%	1.70	65.8%
Buckwalter	Default	42.3%	55.7%	1.70	66.7%
	Ortho Normalized & Dediacritized	43.5%	56.1%	1.70	65.0%
	Diacritized	43.4%	56.4%	1.64	**68.0%**
HSB	Default	42.7%	55.6%	1.66	66.9%
	Ortho Normalized & Dediacritized	43.5%	56.3%	1.69	64.9%
	Diacritized	43.3%	56.7%	1.61	**69.3%**

Table 1: Orthographic encoding results on the test set.

Morphological Scheme	Boundary Marker	Accuracy	±1 Acc	MAE	QWK
WORD	–	39.0%	52.9%	1.72	66.3%
D3TOK	Default (+_/_+)	40.9%	54.0%	1.74	65.4%
	Tatweel	42.0%	54.9%	1.69	**67.4%**
	Space	42.0%	55.2%	1.69	67.0%

Table 2: Morphological encoding results on the test set.

Track	Model	Test				Blind Test			
		Acc	±1 Acc	MAE	QWK	Acc	±1 Acc	MAE	QWK
Strict	PIXEL-English	40.0%	53.0%	1.74	66.5%	41.5%	56.8%	1.6	68.4%
	AraBERTv2 (WORD)	51.1%	65.1%	1.31	76.2%				

Table 3: Strict results on Official and Blind tests vs. AraBERTv2 WORD.

The 'script gap' warrants investigation across additional scripts to determine whether effects reflect Latin-specific advantages or broader visual regularity factors. Future work could explore visual augmentations, different fonts, and document-level readability assessment.

This experiment illustrates how PIXEL can be used to assess the informative potential of specific text manipulations. Tatweel, a native Arabic elongation mark, is presented here merely as an example of a script-internal feature that could be evaluated in this way, with potential relevance for human readers.

Future work could explore PIXEL's ability to capture purely visual cues that affect human reading, such as glyph similarity or diacritic placement. In particular, experiments predicting reading speed could further investigate these effects.

7 Limitations

We tested orthographic variation over the English pretrained PIXEL-base model, giving advantage for Latin characters over Arabic.

We used the default render configuration and it occasionally rendered Arabic script outside of the image.

Acknowledgments

We thank the BAREC shared task organizers for providing the dataset and evaluation framework. We also acknowledge the computational resources provided by our institution's computing cluster.

References

Muhamed Al Khalil, Nizar Habash, and Zhengyang Jiang. 2020. A large-scale leveled readability lexicon for Standard Arabic. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3053–3062, Marseille, France. European Language Resources Association.

Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024. The SAMER Arabic text simplification corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

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

Khalid N. Elmadani, Bashar Alhafni, Hanada Taha, and Nizar Habash. 2025a. BAREC shared task 2025 on Arabic readability assessment. In *Proceedings of the Third Arabic Natural Language Processing Conference*, Suzhou, China. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025b. A large and balanced corpus for fine-grained Arabic readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16376–16400, Vienna, Austria. Association for Computational Linguistics.

Nizar Habash, Abdelhadi Soudi, and Timothy Buckwalter. 2007. On Arabic Transliteration. In Abdelhadi Soudi, Antal Van Den Bosch, and Günter Neumann, editors, *Arabic Computational Morphology*, volume 38, pages 15–22. Springer Netherlands, Dordrecht. Series Title: Text, Speech and Language Technology.

Nizar Habash, Hanada Taha-Thomure, Khalid Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2025. Guidelines for fine-grained sentence-level Arabic readability annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 359–376, Vienna, Austria. Association for Computational Linguistics.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2021. Masked autoencoders are scalable vision learners. *Preprint*, arXiv:2111.06377.

Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2016. Yamama: Yet another multi-dialect arabic morphological analyzer. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: system demonstrations*, pages 223–227.

Juan Liberato, Bashar Alhafni, Muhamed Khalil, and Nizar Habash. 2024. Strategies for Arabic readability modeling. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 55–66, Bangkok, Thailand. Association for Computational Linguistics.

Alberto Muñoz-Ortiz, Verena Blaschke, and Barbara Plank. 2024. Evaluating pixel language models on non-standardized languages. *arXiv preprint arXiv:2412.09084*.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. Camel tools: An open source python toolkit for arabic natural language processing. In *Proceedings of the twelfth language resources and evaluation conference*, pages 7022–7032.

Phillip Rust, Jonas F Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2022. Language modelling with pixels. *arXiv preprint arXiv:2207.06991*.

Hanada Taha-Thomure. 2017. *Arabic Language Text Leveling (معايير هنادا طه لتصنيف مستويات النصوص العربية)*. Educational Book House (دار الكتاب التربوي للنشر والتوزيع).

A Additional Experimental Details

All models were trained using PyTorch 2.5.1 with CUDA 12.4 on two NVIDIA GeForce RTX 3090 GPUs. The rendering pipeline used PangoCairo text renderer. We preprocess all variants with Unicode-normalization and tatweel removal. We used the default architecture composed of 12 Transformer layers, hidden size of 768, 12 attention heads, totaling 86M encoder parameters.

A.1 Training Hyperparameters

Fine-tuning: 86M parameters, sequence length 256, batch size 64, learning rate 5e-05, 7 epochs, dropout 0.1, model selection based on Dev set Cross Entropy loss. Morphological encoding variants were trained on half the batch size and learning rate and on a single GPU.

Code Availability

Our code is available at: https://github.com/bensapirstein/pixel