NeoAraBERT: A Modern Foundation Model for Arabic Embeddings with Diacritics-Aware Tokenization and POS-Targeted Masking

Chadi Abou Chakra[1*], **Hadi Hamoud**[1*], **Osama Rakan Al Mraikhat**[1*†], **Qusai Abu Obaida**[1], **Mohamad Ballout**[2‡], **Fadi A. Zaraket**[1,2‡]
[1]Arab Center for Research and Policy Studies, Doha
[2]American University of Beirut
{cabouchakr,hhamoud,oalmraikhat,qabuobaida,fzaraket}@dohainstitute.edu.qa
{mb185,fz11}@aub.edu.lb

Abstract

We present NeoAraBERT, a state-of-the-art open-source Arabic text-embedding model built on the NeoBERT architecture. We pre-train NeoAraBERT on diverse open-source and internal datasets covering modern standard, classical, and dialectal Arabic. We guided our design choices with Arabic tailored ablation studies including text normalization, light stemming, and diacritics-aware tokenization handling. We also performed more general POS-aware token masking and learning-rate scheduling ablation studies. We benchmarked NeoAraBERT against five top-performing Arabic models on 23 tasks, including a novel synonym-based task, "Muradif", that directly assesses embedding quality with no additional fine-tuning. NeoAraBERT variants (MSA, dialectal, and mixed) rank first in 18 tasks, second in two, third in two, and fourth in one task. They show strong performance on classical and modern standard Arabic, substantial margins of improvement (>7%) in two tasks, and a +2.75% improvement on average across all tasks. Our code and links to checkpoints for our model variants are available on our website: https://acr.ps/neoarabert.

1 Introduction

Recent advances in decoder-based language models have transformed NLP, with models such as Deepseek (Bi et al., 2024), Qwen (Bai et al., 2023; Yang et al., 2025), LLaMA 3 (Dubey et al., 2024), OLMo (Groeneveld et al., 2024), Mistral (Jiang et al., 2023), and Gemma3 (Team et al., 2025) demonstrating strong reasoning and few-shot capabilities. These gains reflect systematic scaling alongside architectural and optimization improvements. Meanwhile, encoder models remain central to retrieval-augmented generation and semantic search (Ram et al., 2023), yet with slower progress.

This gap has recently begun to close with ModernBERT (Warner et al., 2025) and NeoBERT (Breton et al., 2025), which have revitalized the BERT framework through updated architectures, training corpora, and pre-training methodologies. NeoBERT's results indicate that most performance gains arise from improving the pre-trained backbone itself, rather than from sophisticated fine-tuning strategies such as BGE (Xiao et al., 2024) and E5 (Wang et al., 2024). Yet these developments remain largely confined to English, with comparable progress still lacking in Arabic natural language processing.

Current Arabic language encoders like multilingual BERT (Devlin et al., 2019), AraBERT (Antoun et al.) and ARBERT & MARBERT (Abdul-Mageed et al., 2021) remain the predominant choices for Arabic embedding tasks, yet lack the architectural refinements including Rotary Position Embeddings (Su et al., 2024), SwiGLU activations (Shazeer, 2020a), and Pre-RMSNorm (Zhang and Sennrich, 2019) that characterize modern transformer architectures.

We present NeoAraBERT, a modern Arabic encoder that incorporates these components and adopts Arabic-specific training choices. We investigate key design choices for Arabic via ablation studies: (i) we evaluate how normalization techniques and varying degrees of morphological stemming affect performance. (ii) we compare subword and group-level POS-conditioned masking where nouns and verbs take priority. (iii) we transform a diacritized word into the word with no diacritics followed by its diacritics. This *diacritic-aware tokenization* preserves canonical representation via diacritic separation, and compacts the representation of diacritic tokens.

Our training corpus encompasses contemporary Arabic text from diverse sources including academic publications, news media, social media discourse, ensuring representation of both Modern

[*]Equal contribution; authors are listed in random order.
[†]Corresponding author.
[‡]Shared senior authorship.

Findings of the Association for Computational Linguistics: ACL 2026, pages 25952–25968

Standard Arabic (MSA) and regional varieties.

For robust evaluation, we assess model variants on 22 established Arabic NLP tasks and benchmarks (see Section 5). We also introduce a novel Arabic synonym dataset to benchmark semantic similarity. This dataset/benchmark closes a gap in Arabic embedding evaluation resources as it enables direct evaluation of base encoder representations without task-specific fine-tuning.

Our work makes three primary contributions: (i) we develop a modern Arabic encoder architecture that integrates recent innovations and training methodologies; (ii) we conduct systematic ablation studies to identify optimal design choices for Arabic, accounting for properties such as morphological complexity and word and sentence-level structure; and (iii) we release a new evaluation dataset for Arabic embeddings that addresses limitations in current base-encoder evaluation. By modernizing the pre-training foundation for Arabic encoders, we deliver a state-of-the-art Arabic encoder that outperforms existing models on most benchmarks, with especially strong results on MSA and Classical Arabic (CA).

Our ablation studies show consistent gains from text normalization, light stemming, a larger vocabulary size, a higher masking rate with targeted POS-group masking, a cosine-decay learning-rate schedule, and diacritics-aware processing.

Across 23 benchmarks, our released checkpoints win on 18 of them and deliver the best average score, with particularly large gains on Classical Arabic (e.g., APCD Era). $NeoAraBERT_{\text{DA}}$ is the strongest on its dialect-focused benchmark subset.

The rest of this paper reviews related work in Section 2, introduces NeoAraBERT and the data in Section 3, discusses ablation choices and their results in Sections 4 and 7, respectively, and describes the benchmarks and the finetuning settings in Sections 5 and 6. We present and discuss the results in Section 8.

2 Related Work

It has become standard to replicate the BERT release (Devlin et al., 2019) for other languages. Early Arabic encoders included AraBERT (Antoun et al.) (with a second version), ArabicBERT (Safaya et al., 2020) and AraELECTRA (Antoun et al., 2021). Later, ARBERT (Abdul-Mageed et al., 2021) emerged, pretrained exclusively on MSA, with a companion MARBERT pretrained on an additional 1B dialectal tweets. Subsequent versions emerged with extended MSA and news corpora. Subsequent work pushed performance through broader data coverage, multilinguality, and more systematic pretraining and evaluation (Devlin et al., 2019; Goyal et al., 2021; Chung et al., 2020; Liang et al., 2023).

Prior work analyzed factors that matter for Arabic encoders. CAMeLBERT (Inoue et al., 2021) studied the impact of pretraining data size and classical, dialectal, and MSA variation effects on BERT-based encoders. JABER (Ghaddar et al., 2022) explored data quality, model size, and morphology factors. Recently, Swan (Bhatia et al., 2025) introduced Arabic-centric ARBERTv2 and ArMistral-based encoders trained with contrastive objectives over multi-dialect and multi-domain corpora.

Dialect-focused models such as multi-dialect AlcLaM (Ahmed et al., 2024); Moroccan MorrBERT (Moussaoui and El Younnoussi, 2023) and DarijaBERT (Gaanoun et al., 2025); Egyptian EgyBERT (Qarah, 2024a), and SaudiBERT (Qarah, 2024b) emerged. These models primarily improved on data coverage, dialect variation, or training objectives. They kept the underlying BERT-like encoder architecture, with less attention to architectural encoder backbone modernization.

Progress in encoder-only models has been relatively limited compared to the evolution of decoder-based models at the architectural level in recent years. RoBERTa enhanced BERT (Liu et al., 2019). ALBERT (Lan et al., 2019) followed as a lightweight parameter-sharing and embedding factorization variant. SpanBERT (Joshi et al., 2020) modified BERT pre-training objectives to capture span-level linguistic information. ELECTRA (Clark et al., 2020) and DeBERTa (He et al., 2020) introduced more efficient training paradigms and advanced attention mechanisms. Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020) addressed the quadratic self-attention cost with sparse attention patterns to support longer input sequences. Recently, ModernBERT (Warner et al., 2025) and NeoBERT (Breton et al., 2025) incorporated architectural and training innovations from decoder models to revitalize encoder architectures. We adapt NeoBERT innovations to Arabic and conduct Arabic-specific ablation studies to identify effective design choices.

3 Model: NeoAraBERT

We constructed a pre-training corpus encompassing MSA, CA, and Dialectal Arabic (DA). MSA and CA collections span diverse domains, featuring news archives from Assafir, AlArabi, Deutsche Welle (Kahla et al., 2021), and content from SkyNewsArabia and Masrawy. The corpus also incorporates the Abuelkhair corpus (covering Alriyadh, Youm7, Alyaum, Alqabas, Alittihad, Almustaqbal, and Tishreen newspapers and sites) (El-Khair, 2016) and the SANAD dataset (Alkhaleej, Alarabiya, and Akhbarona) (Einea et al., 2019), alongside extracts from BBC Arabic, EuroNews, Aljazeera, CNN Arabic, and RT Arabic. We further included literary, academic, and official texts sourced from Hindawi books, the Open Islamicate Texts Initiative (OpenITI), a private dataset for Quran interpretations, official UN translated transcripts (Ziemski et al., 2016), curricula books covering subjects from Arab countries (Abu Obaida et al., 2025), the Sadeed Tashkeela dataset (Aldallal et al., 2025), and an extract from the OSCAR webcrawl (Ortiz Su'arez et al., 2020; Ortiz Su'arez et al., 2019). We also included available books published by the Arab Center for Research and Policy Studies[1]. Overall, open-source resources constitute 65.95% of the training data, with news-sourced data accounting for 13.67% of the full corpus.

For dialectal Arabic, we utilized Egyptian dialect extracts from OSCAR and a multi-dialect collection from Reddit Pushshift (Baumgartner et al., 2020). The dialectal coverage is enriched by the Currasat collection from SinaLab, which encompasses Levantine (Nayouf et al., 2023; Jarrar et al., 2023b), Egyptian, Gulf, and other regional varieties including Libyan, Iraqi, Sudanese, and Yemeni (Al-Haff et al., 2022). We employed automated quality checking pipelines to identify and clean noisy documents across the entire corpus. We also performed deduplication across all datasets using ONe Instance Only (Pomikálek, 2011). When duplicates were identified based on an approximate 80% similarity threshold, we prioritized retaining the document with higher quality scores. For our ablation studies, we utilized a high-quality subset.

3.1 Normalization and Light-Stemming

We apply a normalization and light-stemming pipeline. We first remove zero-width characters and Tatweel elongation, while preserving diacritics. Our light-stemming approach utilizes the CAMeL Tools framework with disambiguation, implemented in practice through our `fast-disambig` library[2], a Rust-based accelerated replacement for the CAMeL disambiguator (Appendix A.6). We also use a fallback list of 5,000 frequent words, manually segmented by Arabic annotators where CAMeL fails to detect nuanced affixes. We employ the D3 segmentation scheme (third-level segmentation) to split clitics and affixes from the base stem, inserting a special separator token ([+]) to mark segmentation boundaries. Furthermore, we leverage the Universal Dependencies (UD) POS tags assigned to each segmented morpheme. We reconstruct the full diacritization of the original word on the segmented output. Finally, we perform standard normalization for Alif أ and Teh ة variants.

Figure 1: Example of diacritics separation.

3.2 Diacritics-aware Tokenizer

We trained two WordPiece tokenizers on the same data: an undiacritized tokenizer (60k vocabulary size) and a diacritized tokenizer (10k vocabulary size). To ensure diacritics focus for the diacritized tokenizer, we redacted lexical content as follows. We replaced each digit with '1' and each character with Baa ب. Each diacritized or partially diacritized word was undiacritized, normalized with Baa, and then followed by its diacritics sequence, where a dotted circle (◌) represented an omitted diacritic. The diacritized tokenizer learned these diacritic sequences as tokens. We select the top 5k diacritic tokens from it and use them to augment the undiacritized tokenizer. This created our diacritics-aware tokenizer (65k vocabulary size). We trained the models with diacritics via the same separation technique while preserving lexical content (no Baa and digit redactions). Figure 1 shows diacritic separation applied to mrhba (مرحبا).

3.3 Group-Level Masking

Mask groups are constructed by (i) merging subword pieces into whole-word units, (ii) further merging adjacent segments separated by a special stemming marker token ([+]) when the segments share the same POS category, and (iii) attaching

[1] https://acr.ps

[2] https://acr.ps/hBy74Mr

Attribute	AraBERTv2	ARBERTv2 / MARBERTv2	AraModernBERT	NeoAraBERT
Layers	12	12	22	28
Hidden size	768	768	768	768
Attention heads	12	12	12	12
Parameters	∼136M	∼163M	∼149M	∼248M
Activation function	GELU	GELU	GELU	SwiGLU
Positional encoding	Absolute	Absolute	Rotary (RoPE)	Rotary (RoPE)
Normalization	Post-LayerNorm	Post-LayerNorm	Pre-LayerNorm	Pre-RMSNorm
Dataset size	77 GB	243 GB / 128 GB	100 GB	71.2 GB
Tokenizer level	WordPiece	WordPiece	Transtokenization	WordPiece
Vocabulary size	∼64k	100k	50,280	65k
Max sequence length	512	512	8192	1024
Masking scheme	15% random	15% random	30% random	POS-conditioned
Arabic diacritics	Not supported	Not supported	Supported	Supported
Optimizer	Adam (weight decay)	Adam (weight decay)	StableAdamW	AdamW
LR schedule	Warmup + Linear decay	Warmup + Linear decay	Not Reported	Warmup + CosineDecay

Table 1: Comparison of architecture and pre-training configurations for common Arabic encoder baselines.

diacritic-only tokens to the preceding group so that diacritics inherit the same group (and POS category) assignment.

3.4 Pre-training Setup

We use the final pre-training hyperparameters, corresponding to the configuration later selected as **A8**, as summarized in Table 1 and Appendix A.1. The learning rate schedule uses a 2,000-step linear warmup from $10^{-4}\times$ the peak learning rate to a peak of 6×10^{-4}, followed by cosine decay with a minimum learning-rate ratio of 0.1 (i.e., to 6×10^{-5}) until step 266,616 (90% of the total optimization steps), after which the learning rate is held constant for the remaining steps, for a total of 296,240 optimization steps (10 epochs). For the masking rate, we use a POS-conditioned linear schedule with an overall average masking rate of 25.15% described in Appendix A.2 and A.5.

We obtain our first checkpoint, $NeoAraBERT_{\text{MSA}}$, by averaging model weights from epochs two through five. From this checkpoint, we continue training on dialectal data using the same hyperparameters, while resuming the cosine learning-rate decay from epoch five until it reaches approximately 6×10^{-5}. After five epochs (16,365 optimization steps) of dialectal training, we average the weights from the last four epochs to produce our second checkpoint, $NeoAraBERT_{\text{DA}}$. Next, we create an intermediate model by averaging $NeoAraBERT_{\text{MSA}}$ and $NeoAraBERT_{\text{DA}}$. We then train this intermediate model for one additional epoch (21,583 optimization steps) on a high-quality mixture of both MSA and dialectal datasets, using a constant learning rate of 6×10^{-5}. Our final checkpoint, $NeoAraBERT_{\text{Mix}}$, is obtained by averaging the weights of the model after this final epoch with the intermediate averaged model used to initialize it. No improvement was observed after epoch five; we are therefore reporting these results (29.34% overall average masking rate after five epochs).

We train $NeoAraBERT_{\text{MSA}}$ on a single node equipped with 8 NVIDIA H200 GPUs (141 GB each). We use a per-device micro-batch size of 32 and 16 gradient accumulation steps, resulting in an effective global batch size of 4,096. For all ablations and subsequent models, we train on 8 NVIDIA H100 GPUs (80 GB each), using a per-device micro-batch size of 16 and 16 gradient accumulation steps, resulting in an effective global batch size of 2,048. Having described the final model and pre-training pipeline, we now examine the ablation framework used to justify the design choices.

4 Design Choices and Ablation Studies

We select a high-quality subset of our pre-training data containing 1.82B stemmed tokens. Table 5 in Appendix summarizes the successive ablations. Each step modifies its ablation component and retains the settings from the best model so far. We begin with a baseline (**A0**) trained on raw text, with no stemming or normalization. Ablation (**A1**) trains with text normalization including light stemming while keeping WordPiece tokenization, a 30k vocabulary, and a 15% random masking rate fixed; all other settings are inherited from NeoBERT (Breton

et al., 2025).

Ablation **A2** evaluated 30k and 60k vocabulary sizes, following NeoBERT (Breton et al., 2025) and AraBERTv2 (Antoun et al.) vocabulary sizes, respectively. Ablation **A3** compared random masking with 20% and 15% rates (similar to NeoBERT and BERT, respectively).

POS-targeted masking aims to prioritize semantically rich tokens. Masking functional words is generally less informative, whereas masking non-functional words yields stronger learning signals (Yang et al., 2023). To enable targeted masking, we first annotate the data with universal POS tags using CAMeL Tools (Obeid et al., 2020) and compute their frequencies (Table 6). We also shift from token-level to group-level masking (see Section 3.3 for implementation) to cover all constituent tokens of a POS tag.

Ablation **A4** capped the masking probability of any POS tag at 20%. This maintains sufficient context for vocabulary learning and limits the data corruption rate. Without this, random masking would result in a noun masking rate of 27.70% since nouns are dominant. A4 limits it to 20% and increases masking for other less dominant POS categories such as verbs and adjectives. A4 yielded an aggregate masking rate of 14.94% (Table 6).

Ablation **A5** investigated the limits of the prediction rate benefits by raising the cap to 35%. While smaller models suffer with potential corruption, larger models may benefit from higher masking (Wettig et al., 2023). **A5** increased the overall masking rate to 21.8%. Ablation **A6** adopts a dynamic strategy (Yang et al., 2023) that adjusts masking probabilities during training. Based on the loss associated with each POS tag, it automatically detects words the model struggles to predict and improves the value of the masking budget.

Ablation **A7** explores a masking schedule (Ankner et al., 2024; Yang et al., 2023) where rates start high at $\times 1.5$ and decay linearly to a $\times 0.75$ value, using the **A5** cap setting as base. Intuitively, early higher masking injects useful noise and encourages broader exploration, while later lower masking supports refinement and stabilizes learning (Ankner et al., 2024; Yang et al., 2023).

Ablation **A8** introduces diacritics-aware tokenization to incorporate diacritics without compromising canonic semantic representation, and without substantially inflating sequence length. Standard tokenizers treat diacritized words (e.g., مَرحبًا marhaban (hello)) and their undiacritized counterparts (e.g., مرحبا mrhba (hello)) as entirely distinct tokens, effectively fragmenting the semantic space. Note that diacritics are often omitted and readers infer them from context relating both words to the same semantic value.

Diacritic-aware tokenization represents a diacritized word as a sequence of the undiacritized word followed by its diacritics. This approach allows the tokenizer to identify the base word while effectively retaining the diacritic signals when available. These are sometimes instrumental for syntactic, semantic and phonetic disambiguation tasks. We augment the regular 60K tokenizer with 5k secondary diacritic tokens as explained in detail in Section 3.2.

Ablation **A9** adopts a constant learning rate schedule (8×10^{-4} after 2,000 warmup steps) as proposed by ModernBERT (Warner et al., 2025), replacing the cosine learning-rate decay adopted in NeoBERT (Breton et al., 2025). This facilitates continued training without 'cold restart' issues (Ash and Adams, 2020). We evaluate it to ensure it does not degrade performance.

5 Benchmarks

We benchmark our model against state-of-the-art encoders on a diverse suite of Modern Standard Arabic (MSA), dialectal Arabic, and Classical Arabic benchmarks. **Named Entity Recognition (NER)** is evaluated on ANERCorp (Benajiba and Rosso, 2007) and WojoodNER (Jarrar et al., 2022), where the model assigns token-level entity labels (e.g., person, organization, and location). **Part-of-Speech (POS)** tagging uses SALMA (Jarrar et al., 2023a) and UD-Arabic-PADT (Nivre et al., 2020) to predict each token's syntactic category.

Topic classification covers Al Khaleej (Abbas and Smaili, 2005) and ANTCorpusv2 (Chouigui et al., 2017, 2021) where the model predicts the topic given the title, article, or both.

Semantic Textual Similarity (STS) uses Mawdoo3 Q2Q from the NSURL shared task (Seelawi et al., 2019) to predict whether two questions convey the same meaning.

Natural Language Inference (NLI) is evaluated on XNLI (Conneau et al., 2018) (entailment, contradiction, neutral). We additionally evaluate **commonsense reasoning** on ArabicSense (Lamsiyah et al., 2025); in the multiple-choice explanation subtask, the model selects the correct reason

Dataset	Metric	A0	A1	A2	A3	A4	A5	A6
AraSarcasm (Sarc)	F1	65.82	67.86	66.91	**72.44**	69.26	71.00	70.84
AraSarcasm (Sent)	F1	68.59	69.53	**69.85**	68.98	66.77	69.42	69.78
ANTv2 Text	F1	85.55	85.51	85.54	85.70	84.54	86.51	**87.01**
ANTv2 Title	F1	78.92	79.26	79.57	79.99	79.90	**80.81**	80.30
ANERcorp.	μF1	75.46	74.46	76.48	**77.58**	76.67	76.74	76.92
Q2Q(STS)	F1	93.81	94.39	93.85	94.17	93.88	**94.53**	94.49
XNLI	F1	66.52	70.61	70.38	71.51	74.23	74.00	**74.30**
MAWQIF(Sent)	F1	63.82	65.58	64.74	62.71	**67.23**	64.56	64.10
WSD	F1	77.94	77.31	**78.24**	77.52	76.67	77.15	78.10
WoojoodNER	μF1	89.82	90.47	90.19	90.51	90.49	90.49	**90.56**
Muradif (synonyms)	ACC	72.09	75.54	76.52	77.56	77.72	**78.33**	78.27
Average		76.21	77.32	77.48	78.06	77.94	78.50	**78.61**
# of tasks improved		–	8	6	8	3	8	6

Table 2: Performance (F1/μF1/accuracy) on 11 benchmarks for ablations A0–A6. "tasks improved" counts benchmarks that improve over the previous ablation (e.g., A3 vs. A2).

for a nonsensical statement. **Word Sense Disambiguation (WSD)** is evaluated on the Arabic Gloss WSD benchmark (El-Razzaz et al., 2021), where each instance includes an ambiguous word, a candidate gloss, and an example sentence, and the model predicts whether the example matches the gloss. In addition, we evaluate on a **diacritization-specific** benchmark derived from WikiNews-2014 following (Mohamed and Mubarak, 2025). Following (Kharsa et al., 2024), we strip diacritics from each sentence and train the model to recover them. We formulate this task as token-level classification over relative-position (RP) labels, and construct leakage-aware splits by grouping identical undiacritized sentences.

In addition, we evaluate our model on **Classical Arabic** using tasks such as predicting the emotion in a poem (Shahriar et al., 2023). We also evaluate the models on APCD (Yousef et al., 2019), which contains around 1.8 million samples, for identifying the meter and the era of a given verse. APCD includes 23 meter classes and 12 era classes.

Dialectal evaluation includes **sentiment** and **sarcasm** on ArSarcasm (Farha and Magdy, 2020) and MAWQIF (Alturayeif et al., 2022), **stance detection** on MAWQIF, and **dialect identification** on Arabic Dialects (El-Haj et al., 2018) (Egyptian, North African, Gulf, Levantine, and MSA).

5.1 Muradif Benchmark

Semantic equivalence is fundamental to semantic search and RAG applications where such models remain central. We introduce "Muradif"[3] مرادف (synonym), a synonym-based benchmark that tests whether distinct lexical items with similar or identical semantics are mapped to proximal embedding representations. We built Muradif using the Arabic ontology dataset from SinaLab (Jarrar, 2021; Jarrar and Amayreh, 2019) by picking synonym sets that have at least two words or phrases and at least one sentence context. Muradif consists of 38,554 instances. An instance in Muradif consists of a sentence representing a context, and three candidates to be inserted in the sentence. The candidates are an anchor picked randomly from the synonym set, a synonym to that anchor, and a word or phrase that is irrelevant to the anchor and its synonym. We form three sentences by substituting the three candidates in the subject sentence. Then we compare the aggregated cosine distances of the resulting sentences. We aggregate by computing the mean-pooled embeddings of the sentence. A positive is counted if the anchor to synonym similarity is higher than the anchor to irrelevant similarity; otherwise, the result is negative. This aims to check semantic consistency in embedding space produced by the model with no additional fine-tuning.

6 Fine-tuning Settings

We fine-tuned our model for the benchmark tasks and all baselines using the same training pipeline and hyperparameters. All experiments were run with a maximum sequence length of 512 and a batch size of 32, and we fine-tuned for 5 epochs using AdamW with a learning rate of 2e-5. We used a linear learning-rate schedule with a warmup ratio of 10% (computed from the total number of

[3] https://acr.ps/muradif

Dataset	Metric	**A4**	**A5**	**A7**	**A8**	**A9**
AraSarcasm (Sarc)	F1	72.10	70.88	71.18	**72.83**	71.09
AraSarcasm (Sent)	F1	68.68	**70.35**	69.65	69.83	68.33
ANTv2 Text	F1	86.39	86.87	**87.65**	87.37	85.92
ANTv2 Title	F1	80.59	**81.66**	79.99	81.61	79.77
ANERcorp.	μF1	78.61	78.52	78.06	**78.72**	77.12
Q2Q(STS)	F1	94.74	94.47	94.71	**94.78**	93.65
XNLI	F1	**76.71**	76.23	74.38	75.53	76.03
MAWQIF(Sent)	F1	64.34	65.82	**66.64**	65.68	63.94
WSD	μF1	80.83	77.59	80.36	**81.21**	78.52
SALMA(POS)	μF1	96.23	96.42	96.70	**96.83**	95.68
ud (POS)	F1	96.65	96.65	**96.83**	96.78	96.28
ArabicSense(reason)	F1	97.05	**97.87**	**97.87**	97.75	93.48
ArabicSense(nli)	F1	**99.88**	**99.88**	99.64	99.76	99.64
Arabic Dialects	F1	72.04	75.23	**75.33**	75.02	73.03
WoojoodNER	μF1	91.18	91.21	91.48	91.48	**91.71**
Muradif (synonyms)	ACC	80.22	81.32	81.53	83.55	**83.72**
APCD_meter	F1	79.10	79.10	79.67	**84.07**	83.54
APCD_era	F1	29.54	30.10	30.70	**48.76**	48.10
Average		80.27	80.57	80.69	**82.31**	81.09
# of tasks improved		–	10	12	12	3

Table 3: Performance (F1, μF1, or accuracy) across evaluation benchmarks for ablations A4–A9 under different masking and diacritics-related configurations, trained for 3 epochs. "# of tasks improved" counts how many benchmarks improve relative to the immediately preceding ablation (e.g., A7 vs. A5).

training steps) and applied gradient clipping with a maximum norm of 1.0 for stability. Depending on the task, we report F1, μF1, or accuracy. We used the official train/dev/test splits when available; otherwise, we created our own 80/10/10 splits. We select the checkpoint with the best validation score and report test performance using that checkpoint. For the large APCD benchmark (around 1.8M samples), we fine-tuned for only 3 epochs with a maximum sequence length of 128. XNLI, with 400K samples, was also limited to a maximum sequence length of 128, and trained for 5 epochs.

7 Ablation Results

Ablations **A0**-**A6** trained for one epoch on 1.82B stemmed tokens from our highest-quality MSA and Classical Arabic subset. We benchmarked each checkpoint on 11 tasks as reported in Table 2.

A1 shows that text normalization with light stemming improves performance on most tasks. Increasing vocabulary size (**A2**) yields a consistent gain. Increasing the random masking rate from 15% to 20% improves performance (**A3**) as the model benefits from a stronger learning signal.

POS-targeted masking yields further gains. Capped masking rate of dominant POS groups reallocates masks to other POS categories and improves performance even with an overall masking rate close to 15% (**A4** vs. **A2**). Emphasis on content words (verbs, nouns, and adjectives) allows a higher overall masking rate and yields the strongest results among masking settings (**A5**). The dynamic strategy **A6** slightly improves results; however, it significantly increases training time.

Based on the results, we retain **A4** and **A5** and retrain their checkpoints for three epochs. We then evaluate them on 16 tasks to select the best configuration as reported in Table 3. We chose to build **A7** based on **A5** due to its slight performance advantage. Additionally, we anticipate that **A5**'s higher masking rate will provide a better learning environment, particularly during extended training where the model has sufficient epochs to overcome the increased data corruption. The addition of diacritics in **A8** led to a significant performance boost. In contrast, **A9** demonstrated that switching to a constant learning rate causes a marked deterioration in results compared to the cosine decay used in **A8**. We select **A8** for full pre-training based on the results in Table 3 and our theoretical expectations.

To isolate the effect of the diacritics-aware tokenizer, we compare the ablations before its introduction (A4, A5, and A7) with those after its introduction (A8 and A9). On APCD Meter and APCD ERA, two fully or highly diacritized Classical Arabic benchmarks, we observe substantial improvements. In particular, comparing Ablation A7 with Ablation A8, where the only difference

Benchmark			Mix	MSA	DA	M1	M2	M3	M4	M5	Rank
AraSarcasm (Sarc)	F1	DA	73.48	74.19	75.24	74.42	74.97	**76.48**	72.27	71.91	2
AraSarcasm (Sent)	F1	DA	73.36	70.47	73.09	73.68	71.29	**73.89**	70.78	72.92	3
MAWQIF (Stance)	F1	DA	67.64	66.81	**70.94**	65.74	65.13	70.10	65.92	66.35	1
MAWQIF (Sent)	F1	DA	69.23	66.09	**69.56**	68.54	64.73	69.54	65.66	65.84	1
Arabic Dialects	F1	DA	**79.18**	75.32	78.20	77.36	79.02	78.39	75.89	76.96	1
ANTv2 Text	F1	MSA	88.31	**88.71**	88.47	88.13	88.45	87.97	87.16	88.09	1
ANTv2 Title	F1	MSA	81.85	**82.97**	82.65	82.48	82.35	82.37	81.27	81.62	1
ANTv2 Text + Title	F1	MSA	87.71	**88.70**	88.05	88.17	88.30	87.91	87.44	87.34	1
Al Khaleej	F1	MSA	95.39	**95.42**	94.79	94.89	95.23	95.18	94.81	94.91	1
ANERcorp.	μF1	MSA	81.42	80.12	80.76	82.10	**82.23**	78.88	73.41	80.83	3
WoojoodNER	μF1	MSA	91.36	**91.90**	90.93	90.91	84.72	89.12	91.08	88.43	1
Q2Q(STS)	F1	MSA	95.26	95.45	94.75	**96.29**	95.48	95.21	96.01	95.04	4
XNLI	F1	MSA	**80.84**	80.08	80.66	79.28	76.40	74.90	78.53	73.36	1
Woojood_hadath	F1	MSA	89.67	90.53	89.57	90.53	89.69	90.10	**92.37**	89.00	2
ArabicSense(reason)	F1	MSA	**98.82**	98.23	98.23	97.76	96.46	96.35	92.80	96.57	1
SALMA(POS)	μF1	MSA	**97.26**	97.02	96.67	94.59	97.04	96.10	93.70	96.57	1
ud (POS)	μF1	MSA	**97.07**	96.89	96.84	95.97	96.85	96.56	95.77	95.83	1
WSD	F1	MSA	**83.51**	82.18	81.57	83.48	80.76	79.74	79.74	79.98	1
Muradif (synonyms)	ACC	MSA	**87.03**	86.32	82.64	64.56	73.41	67.15	77.33	67.52	1
Wiki_news (diacritics)	ACC	MSA	**94.84**	94.74	94.21	89.13	89.38	84.50	89.42	94.35	1
APCD_meter	F1	CA	**85.34**	**85.34**	84.94	77.69	77.31	77.61	83.09	77.70	1
APCD_era	F1	CA	**53.71**	52.97	50.85	26.70	25.59	28.26	46.21	26.21	1
Poem_emotion	F1	CA	74.87	75.54	**75.60**	74.82	72.25	74.11	73.36	73.50	1
Average			**83.79**	83.30	83.44	80.75	80.31	80.45	81.04	80.04	**18**

Table 4: Comparison of $NeoAraBERT_{\text{Mix}}$, $NeoAraBERT_{\text{MSA}}$, and $NeoAraBERT_{\text{DA}}$ (Mix, MSA, and DA, respectively) with M1=AraBERTv2 (Antoun et al.), M2=ARBERTv2, M3=MARBERTv2 (Abdul-Mageed et al., 2021), M4=AraModernBERT (NAMAA, 2025), and M5=CAMeLBERT-mix (Inoue et al., 2021) across 23 tasks.

between the two checkpoints is the inclusion of diacritics, A7 achieves 79.67% on APCD Meter and 30.70% on APCD ERA, while A8 reaches 84.07% and 48.76%, respectively. Because diacritics are the only difference between these two settings, and because both benchmarks are heavily diacritized, these gains can be directly attributed to the inclusion of diacritics.

8 Results and Analysis

In Table 4, we report results for our three checkpoints, $NeoAraBERT_{\text{Mix}}$, $NeoAraBERT_{\text{MSA}}$, and $NeoAraBERT_{\text{DA}}$, and compare them with baseline models across three benchmark categories: Dialectal Arabic (DA), Modern Standard Arabic (MSA), and Classical Arabic (CA). The benchmarking results show that the proposed model is state of the art, outperforming the other models by a large margin. The ranks (1-6) indicate the best-performing NeoAraBERT checkpoint relative to the five baseline models.

On the dialectal benchmarks, our checkpoints win three out of five tasks, with two wins from $NeoAraBERT_{\text{DA}}$ and one from $NeoAraBERT_{\text{Mix}}$, while MARBERTv2 wins the remaining two.

On the MSA benchmarks, our MSA checkpoint outperforms AraBERTv2 and ARBERTv2 on 12 and 13 of the 15 tasks, respectively. It also outperforms MARBERTv2 on all 15 tasks and surpasses AraModernBERT on 14 of the 15 tasks. We attribute these gains to a combination of architectural updates and the quality of our MSA pre-training data, especially given that our model is pre-trained for fewer epochs and on less data than AraBERTv2, ARBERTv2, MARBERTv2, and AraModernBERT (Table 1).

In semantic equivalence, which matters most for RAG applications, we observe large gains on Muradif, where $NeoAraBERT_{\text{Mix}}$ achieves 87.03%,

compared to 64.56% (AraBERTv2), 73.41% (ARBERTv2), 67.15% (MARBERTv2), and 67.52% (CAMeLBERT-mix). The closest baseline is AraModernBERT at 77.33%, still about 10 points behind. This highlights a clear strength on synonym-based evaluation of embedding quality.

Our models also perform strongly on natural language inference and reasoning. On XNLI, $NeoAraBERT_{\text{Mix}}$ exceeds the best baseline (AraBERTv2) by 1.5 points. In addition, $NeoAraBERT_{\text{Mix}}$ outperforms other baselines on WSD and POS tagging. $NeoAraBERT_{\text{MSA}}$ dominates topic classification, winning all four tasks across Al Khaleej and ANTv2. For NER, our model achieves the best result on WoojoodNER, while ARBERTv2 is strongest on ANER.

Our model is particularly strong on Classical Arabic, showing a substantial performance gap relative to competing models. Across the three Classical Arabic benchmarks, all of our checkpoints outperform the baselines, and the largest margins in Table 4 appear in this category. For example, on APCD Era (predicting the era of a verse), $NeoAraBERT_{\text{Mix}}$ achieves 53.71% F1, while AraBERTv2, ARBERTv2, and MARBERTv2 score 26.70%, 25.59%, and 28.26%, respectively. The closest baseline is AraModernBERT at 46.21%. A similar trend holds for meter prediction, where $NeoAraBERT_{\text{Mix}}$ scores 85.34% versus 77.69% (AraBERTv2), 77.31% (ARBERTv2), and 77.61% (MARBERTv2). These gains are primarily attributable to our explicit handling of diacritics, a capability absent in AraBERTv2 and ARBERTv2. Our ablation study, particularly the comparison between A7 and A8, confirms that diacritic modeling is the primary contributor to these improvements (see Section 7). Differences in the quality and composition of the pre-training data may have further amplified these gains.

Our model also performs strongly on the diacritization-specific benchmark, highlighting the effectiveness of our diacritic-aware tokenization scheme. On WikiNews-2014, it achieves the best result, with 94.84%, surpassing AraBERTv2, ARBERTv2, MARBERTv2, CAMeLBERT-Mix, and AraModernBERT, as reported in Table 4.

Finally, we observe a trade-off between dialect specialization and MSA performance. Continued training on dialectal data alone can reduce performance on MSA benchmarks, as reflected by $NeoAraBERT_{\text{DA}}$ not winning any MSA tasks. In contrast, $NeoAraBERT_{\text{Mix}}$, which combines quality dialectal and MSA datasets with an additional epoch of training, yields a more balanced model that performs strongly across MSA, Classical Arabic, and dialectal benchmarks.

9 Conclusion

In this paper, we release NeoAraBERT, a state-of-the-art Arabic encoder model based on the NeoBERT architecture. The model outperforms existing baselines on most benchmarks and particularly excels in Modern Standard Arabic and Classical Arabic. Developing the model involved a thorough set of ablation studies, including a dedicated analysis of masking strategies and POS-targeted masking, as well as several components tailored to Arabic. These ablations provide practical insights for the community on effective pre-training choices. Overall, our models achieve the best performance on 18 out of 23 tasks, with an average score of 83.79% ($NeoAraBERT_{\text{Mix}}$) across all tasks, compared to 81.04% for the closest competing model. We also introduce a novel synonym-based benchmark (Muradif) for evaluating embedding quality without task-specific fine-tuning.

In future work, we plan to extend the context length to 4,096 tokens to better support long-document tasks. We also plan to explore multilingual NeoBERT-style pre-training that includes Arabic, enabling stronger cross-lingual transfer while retaining Arabic-specific modeling choices.

Limitations

Even though our dialectal Arabic data covers a wide range of Arabic-speaking countries, gaps remain for some regions due to data limitations. In addition, while our diacritics-aware tokenization and stemming generally improve performance, especially on Classical Arabic, they increase token counts and effectively shorten the usable context length. Additionally, we could not evaluate our model on ArabicMTEB (Bhatia et al., 2025) because we did not have access to their GitHub repository and datasets. Finally, we annotate our pre-training corpus with POS tags using CAMeL Tools (Obeid et al., 2020), which may introduce tagging errors compared to human annotation.

Ethics Statement

Designed for Modern Standard Arabic (MSA) and dialects, this embedding model was trained on a mix of internal proprietary data and public archives.

We cannot release the dataset due to privacy restrictions and content volatility. Despite high benchmark performance, this broad training data may introduce societal biases. Consequently, users must navigate limitations regarding linguistic fairness and religious sensitivity.

Acknowledgments

We would like to acknowledge Ahmad Talal Salman from Assafir and Professor Amer Abdo Mouawad from the American University of Beirut for sharing Assafir data, which was instrumental to the work presented in this paper.

References

Mourad Abbas and Kamel Smaili. 2005. Comparison of topic identification methods for arabic language. In *Proceedings of International Conference on Recent Advances in Natural Language Processing, RANLP*, pages 14–17.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and 1 others. 2021. Arbert & marbert: Deep bidirectional transformers for arabic. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 7088–7105.

Qusai Abu Obaida, Digital Research Unit Arab Center for Research, and Policy Studies (U4RASD). 2025. U4rasd/curriculum_books_sft. Hugging Face Datasets. Version: main (commit 899b1f4). Accessed: 2026-01-05.

Murtadha Ahmed, Saghir Alfasly, Bo Wen, Jamal Addeen, Mohammed Ahmed, and Yunfeng Liu. 2024. Alclam: Arabic dialect language model. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 153–159.

Karim Al-Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a Levantine corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 769–778, Marseille, France. European Language Resources Association.

Zeina Aldallal, Sara Chrouf, Khalil Hennara, Mohamed Motaism Hamed, Muhammad Hreden, and Safwan AlModhayan. 2025. Sadeed: Advancing arabic diacritization through small language model. *Preprint*, arXiv:2504.21635.

Nora Saleh Alturayeif, Hamzah Abdullah Luqman, and Moataz Aly Kamaleldin Ahmed. 2022. Mawqif: A multi-label arabic dataset for target-specific stance detection. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 174–184.

Zachary Ankner, Naomi Saphra, Davis Blalock, Jonathan Frankle, and Matthew Leavitt. 2024. Dynamic masking rate schedules for mlm pretraining. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 477–487.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Araelectra: Pre-training text discriminators for arabic language understanding. In *Proceedings of the sixth arabic natural language processing workshop*, pages 191–195.

Jordan Ash and Ryan P Adams. 2020. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. *CoRR*, abs/2001.08435.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yassine Benajiba and Paolo Rosso. 2007. Anersys 2.0: Conquering the ner task for the arabic language by combining the maximum entropy with pos-tag information. In *IICAI*, pages 1814–1823.

Gagan Bhatia, Abdellah El Mekki, Fakhraddin Alwajih, Muhammad Abdul-Mageed, and 1 others. 2025. Swan and arabicmteb: Dialect-aware, arabic-centric, cross-lingual, and cross-cultural embedding models and benchmarks. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4654–4670.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Lola Le Breton, Quentin Fournier, John Xavier Morris, Mariam El Mezouar, and Sarath Chandar. 2025. NeoBERT: A next generation BERT. *Transactions on Machine Learning Research*. Reproducibility Certification.

Amina Chouigui, Oussama Ben Khiroun, and Bilel Elayeb. 2021. An arabic multi-source news corpus: experimenting on single-document extractive summarization. *Arabian Journal for Science and Engineering*, 46(4):3925–3938.

Amina Chouigui, Oussama Ben Khiroun, and Bilel Elayeb. 2017. Ant corpus: An arabic news text collection for textual classification. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 135–142. IEEE.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2020. Rethinking embedding coupling in pre-trained language models. *arXiv preprint arXiv:2010.12821*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2475–2485.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Omar Einea, Ashraf Elnagar, and Ridhwan Al Debsi. 2019. Sanad: Single-label arabic news articles dataset for automatic text categorization. *Data in Brief*, 25:104076.

Mahmoud El-Haj, Paul Rayson, and Mariam Aboelezz. 2018. Arabic dialect identification in the context of bivalency and code-switching. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan.*, pages 3622–3627. European Language Resources Association.

Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *CoRR*, abs/1611.04033.

Mohammed El-Razzaz, Mohamed Waleed Fakhr, and Fahima A Maghraby. 2021. Arabic gloss wsd using bert. *Applied Sciences*, 11(6):2567.

Ibrahim Abu Farha and Walid Magdy. 2020. From arabic sentiment analysis to sarcasm detection: The arsarcasm dataset. In *The 4th Workshop on Open-Source Arabic Corpora and Processing Tools*, pages 32–39. European Language Resources Association (ELRA).

Kamel Gaanoun, Abdou Mohamed Naira, Anass Allak, and Imade Benelallam. 2025. Darijabert: a step forward in nlp for the written moroccan dialect. *International Journal of Data Science and Analytics*, 20(2):917–929.

Abbas Ghaddar, Yimeng Wu, Sunyam Bagga, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Xinyu Duan, Zhefeng Wang, and 1 others. 2022. Revisiting pre-trained language models and their evaluation for arabic natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3135–3151.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. *arXiv preprint arXiv:2105.00572*.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, and 1 others. 2024. Olmo: Accelerating the science of language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 15789–15809.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104.

Mustafa Jarrar. 2021. The arabic ontology – an arabic wordnet with ontologically clean content. *Applied Ontology*, 16(1):1–26.

Mustafa Jarrar and Hamzeh Amayreh. 2019. An arabic-multilingual database with a lexicographic search engine. In *Natural Language Processing and Information Systems*, pages 234–246, Cham. Springer International Publishing.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested Arabic named entity corpus and recognition using BERT. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3626–3636, Marseille, France. European Language Resources Association.

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammed Khalilia. 2023a. Salma: Arabic sense-annotated corpus and wsd benchmarks. *arXiv preprint arXiv:2310.19029*.

Mustafa Jarrar, Fadi A Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Wählisch. 2023b. Lisan: Yemeni, iraqi, libyan, and sudanese arabic dialect corpora with morphological annotations.

In *2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

Mram Kahla, Zijian Győző Yang, and Attila Novák. 2021. Cross-lingual fine-tuning for abstractive Arabic text summarization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 655–663, Held Online. INCOMA Ltd.

Ruba Kharsa, Ashraf Elnagar, and Sane Yagi. 2024. Bert-based arabic diacritization: A state-of-the-art approach for improving text accuracy and pronunciation. *Expert Systems with Applications*, 248:123416.

Salima Lamsiyah, Kamyar Zeinalipour, Matthias Brust, Marco Maggini, Pascal Bouvry, Christoph Schommer, and 1 others. 2025. Arabicsense: A benchmark for evaluating commonsense reasoning in arabic with large language models. In *Proceedings of the 4th Workshop on Arabic Corpus Linguistics (WACL-4)*, pages 1–11.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. Xlm-v: Overcoming the vocabulary bottleneck in multilingual masked language models. *arXiv preprint arXiv:2301.10472*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Abubakr Mohamed and Hamdy Mubarak. 2025. Advancing arabic diacritization: Improved datasets, benchmarking, and state-of-the-art models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16718–16730.

Otman Moussaoui and Yacine El Younnoussi. 2023. Pre-training two bert-like models for moroccan dialect: Morroberta and morrbert. In *MENDEL*, volume 29, pages 55–61.

NAMAA. 2025. Aramodernbert: Advanced arabic language model through trans-tokenization and modernbert architecture. https://huggingface.co/NAMAA-Space/AraModernBert-Base-V1.0. Accessed: 2025-03-02.

Amal Nayouf, Tymaa Hammouda, Mustafa Jarrar, Fadi Zaraket, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian Arabic dialects with morphological annotations. In *Proceedings of ArabicNLP 2023*, pages 12–23, Singapore (Hybrid). Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Jan Hajic, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the twelfth language resources and evaluation conference*, pages 4034–4043.

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 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Pedro Javier Ortiz Su'arez, Laurent Romary, and Benoit Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Pedro Javier Ortiz Su'arez, Benoit Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut f"ur Deutsche Sprache.

Jan Pomikálek. 2011. *Removing boilerplate and duplicate content from web corpora*. Ph.D. thesis, Masaryk university, Faculty of informatics, Brno, Czech Republic.

Faisal Qarah. 2024a. Egybert: A large language model pretrained on egyptian dialect corpora. *arXiv preprint arXiv:2408.03524*.

Faisal Qarah. 2024b. Saudibert: A large language model pretrained on saudi dialect corpora. *arXiv preprint arXiv:2405.06239*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav

Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. *arXiv preprint arXiv:2007.13184*.

Haitham Seelawi, Ahmad Mustafa, Hesham Al-Bataineh, Wael Farhan, and Hussein T Al-Natsheh. 2019. Nsurl-2019 shared task 8: Semantic question similarity in arabic. *arXiv preprint arXiv:1909.09691*.

Sakib Shahriar, Noora Al Roken, and Imran Zualkernan. 2023. Classification of arabic poetry emotions using deep learning. *Computers*, 12(5):89.

Noam Shazeer. 2020a. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Noam Shazeer. 2020b. Glu variants improve transformer. *ArXiv*, abs/2002.05202.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, Dubrovnik, Croatia. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Dongjie Yang, Zhuosheng Zhang, and Hai Zhao. 2023. Learning better masking for better language model pre-training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7255–7267.

Waleed A Yousef, Omar M Ibrahime, Taha M Madbouly, and Moustafa A Mahmoud. 2019. Learning meters of arabic and english poems with recurrent neural networks: a step forward for language understanding and synthesis. *arXiv preprint arXiv:1905.05700*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in neural information processing systems*, 32.

Micha
l Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

A Appendix

A.1 Hyperparameters

We train our models from scratch, adopting the modern architectural enhancements used in NeoBERT (Breton et al., 2025). Specifically, we incorporate Rotary Positional Embeddings (RoPE) (Su et al., 2023), Swish Gated Linear Unit (SwiGLU) activation functions (Shazeer, 2020b), and Pre-norm Root Mean Square Normalization (Pre-RMSNorm) (Zhang and Sennrich, 2019) with $\epsilon = 10^{-5}$. The architecture consists of 28 hidden layers, a model hidden size $d_{\text{model}} = 768$, and 12 attention heads. We use the AdamW optimizer with weight decay 0.1, $\epsilon = 10^{-8}$, $\beta_1 = 0.9$, and $\beta_2 = 0.95$. Following (Shazeer, 2020b), for SwiGLU-based Feed-Forward Network (FFN) layers we set $d_{ff} = 2048$ (i.e., $\frac{2}{3}$ of 3072) to keep parameter count and compute comparable to a standard FFN, since SwiGLU-based FFNs use three weight matrices instead of two.

A.2 POS-Conditioned Linear Schedule

For each POS category c, we select each candidate mask group independently with a Bernoulli trial. In our implementation, the decay schedule is applied using worker-local masking steps. Let W denote the number of data-loader workers per training process, let A denote the number of gradient accumulation steps, and let T denote the number of optimization steps. Each worker increments its local masking step counter once per micro-batch, i.e., once per call to the masking collator. We therefore set the schedule length S to the approximate number of micro-batches per worker:

$$S \approx \left\lfloor \frac{T \cdot A}{W} \right\rfloor \tag{1}$$

The masking probability at local step $s \in \{0, \ldots, S-1\}$ is defined as:

$$P_c(s) = P_c^{\text{init}} + \frac{s}{S-1}\left(P_c^{\text{final}} - P_c^{\text{init}}\right) \tag{2}$$

where P_c^{init} and P_c^{final} denote the initial and final masking probabilities for category c. Under this linear schedule, the average masking probability for category c over training is:

$$\bar{P}_c = \frac{P_c^{\text{init}} + P_c^{\text{final}}}{2}. \tag{3}$$

Since POS categories occur with different frequencies in the corpus, we report an overall average masking rate as a frequency-weighted average across categories. Let f_c denote the frequency of POS category c in the training data, normalized such that $\sum_{c \in C} f_c = 1$, where C is the set of POS categories. We define the overall initial and final masking rates as:

$$P_i^{\text{overall}} = \sum_{c \in C} f_c \, P_c^{\text{init}} \tag{4}$$

$$P_f^{\text{overall}} = \sum_{c \in C} f_c \, P_c^{\text{final}} \tag{5}$$

For the linear schedule, the overall average masking rate is:

$$\bar{M} = \sum_{c \in C} f_c \, \bar{P}_c = \frac{P_i^{\text{overall}} + P_f^{\text{overall}}}{2} \tag{6}$$

Because selection operates on mask groups, which may span variable numbers of subword tokens, $\bar{M}$ corresponds to an expected *group-level* masking rate. Table 7 reports f_c, P_c^{init}, and P_c^{final} for each POS category and the corresponding overall rates P_i^{overall}, P_f^{overall}, and $\bar{M}$.

A.3 Ablation Setup and Successive Modifications

We summarize the modifications across ablations in Table 5, where each row changes exactly one component relative to the previous setting.

Ablation	Modification	Before	After
A1	Text normalization (incl. light stemming)	No normalization	Normalize text (incl. light stemming)
A2	Vocabulary size	30k WordPiece	60k WordPiece
A3	Random masking rate	15% random masking	20% random masking
A4	POS-targeted masking (capped)	Random masking (20%)	Cap each POS group at 20% Upweight verbs and adjectives Overall masking 14.94% Group-level masking
A5	POS-targeted masking (content-focused)	POS-targeted (capped; *A4*)	Increase masking for all POS groups Overall masking 21.8%
A6	POS-targeted masking (loss-driven)	Manually set POS masking rates (*A5*)	Dynamic POS masking using per-POS loss signals(functional vs content words)
A7	Masking-rate schedule (dynamic scaling)	Fixed POS-targeted rates (*A5*)	Linear decay scaling: $\times 1.5$ (early) $\rightarrow$ $\times 0.75$ (late). Based on *A5* distribution
A8	Tokenizer + diacritics handling	WordPiece (60k) Normalize diacritics	Diacritics-aware tokenizer: 60k undiacritized + 5k diacritic (65k total)
A9	Learning-rate schedule	Cosine decay (with warmup)	Warmup 2000 steps, then constant LR 8×10^{-4}

Table 5: Modifications between successive ablations. The initial *A0* baseline is trained on raw text without stemming/normalization; all other settings follow NeoBERT (Breton et al., 2025) unless stated otherwise.

A.4 POS Distribution and Targeted Masking

Table 6 reports the empirical POS-tag distribution f_c in our ablation corpus and the corresponding per-tag masking rates used by the POS-targeted strategies. Compared to the capped setting in *A4*, *A5* increases masking on content-bearing categories (e.g., NOUN/PROPN/VERB/ADJ), leading to a higher overall masking rate.

POS Tag (c)	f_c **(%)**	**A4**	**A5**
ADJ	5.73	20%	30%
PRON	9.15	10%	10%
NOUN	27.70	20%	35%
INTJ	0.02	5%	15%
PART	2.26	10%	10%
AUX	0.45	5%	5%
ADP	11.52	10%	10%
CCONJ	6.59	10%	10%
SCONJ	3.18	10%	10%
PUNCT	11.67	10%	10%
NUM	1.22	15%	20%
PROPN	6.38	20%	35%
VERB	10.70	20%	30%
X	0.70	5%	5%
DET	2.27	5%	5%
ADV	0.48	10%	20%
Masking Rate	100.00%	14.94%	21.8%

Table 6: Percentages of part-of-speech tags in the ablation dataset and targeted masking rates for each tag in ablations A4 and A5.

A.5 Dynamic Masking Schedule

We apply a linear masking schedule that interpolates between an initial POS-specific masking probability P_i and a final probability P_f over the course of training. Table 7 lists P_i and P_f for each POS category, together with the corresponding mean value $\bar{P}_c$ under linear interpolation. The bottom row reports the corpus-weighted overall initial and final masking probabilities, P_i^{overall} and P_f^{overall}, as well as the overall mean masking rate $\bar{M}$. Also, we report the per-class masking probability and corpus-weighted mean at 50% of training steps, corresponding to epoch 5, as $P_c^{(5)}$ and $\bar{M}_5$.

POS Tag (c)	f_c (%)	P_i	P_f	$\bar{P}_c$	$P_c^{(5)}$
NOUN	27.85	0.5250	0.2625	0.3938	0.4594
PROPN	7.95	0.5250	0.2625	0.3938	0.4594
VERB	10.81	0.4500	0.2250	0.3375	0.3938
ADJ	6.06	0.4500	0.2250	0.3375	0.3938
NUM	1.17	0.3000	0.1500	0.2250	0.2625
ADV	0.42	0.3000	0.1500	0.2250	0.2625
INTJ	0.02	0.2250	0.1125	0.1688	0.1969
ADP	11.30	0.1500	0.0750	0.1125	0.1312
PUNCT	10.41	0.1500	0.0750	0.1125	0.1312
PRON	9.31	0.1500	0.0750	0.1125	0.1312
CCONJ	6.65	0.1500	0.0750	0.1125	0.1312
SCONJ	3.17	0.1500	0.0750	0.1125	0.1312
PART	2.25	0.1500	0.0750	0.1125	0.1312
DET	1.23	0.0750	0.0375	0.0562	0.0656
X	1.01	0.0750	0.0375	0.0562	0.0656
AUX	0.39	0.0750	0.0375	0.0562	0.0656
	$\sum_{c \in C} f_c = 100.00$	$P_i^{\text{overall}} = 0.3353$	$P_f^{\text{overall}} = 0.1676$	$\bar{M} = 0.2515$	$\bar{M}_5 = 0.2934$

Table 7: Linear masking schedule by POS tag category.

A.6 Optimized Stemmer based on CAMeL

To make large-scale Arabic preprocessing practical, and to bundle an efficient stemmer with the model, we implemented `fast-disambig`, a Rust-based Arabic morphological disambiguation and stemming library with Python bindings. The library is designed as a drop-in replacement for the CAMeL Tools MLE disambiguator, with an exposed Analyzer that could be used with CAMeL's BERT unfactored disambiguator, while preserving the stemming behavior needed by our normalization and light-stemming pipeline. In our setup, this reduced preprocessing time for corpus preparation and ablation experiments. On an Apple M1 Max, benchmarked on the Hindawi Books dataset, fast-disambig processes a single text in 38 ms, compared with 340 ms for CAMeL Tools, a 9× speedup. It also processes 491 book chapters (7.1 million characters) in 19 s, compared with 19 min 26 s for CAMeL Tools, a 61× speedup.

Workload	`fast-disambig`	CAMeL Tools	Speedup
Single text	38ms	340ms	9×
491 chapters (7.1M chars)	19s	19m 26s	61×

Table 8: Preprocessing speed comparison from the `fast-disambig` README benchmark on the Hindawi Books dataset.