Lemmatization as a Classification Task: Results from Arabic across Multiple Genres

Mostafa Saeed and **Nizar Habash**
Computational Approaches to Modeling Language (CAMeL) Lab
New York University Abu Dhabi
{mostafa.saeed,nizar.habash}@nyu.edu

Abstract

Lemmatization is crucial for NLP tasks in morphologically rich languages with ambiguous orthography like Arabic, but existing tools face challenges due to inconsistent standards and limited genre coverage. This paper introduces two novel approaches that frame lemmatization as classification into a **Lemma-POS-Gloss (LPG)** tagset, leveraging machine translation and semantic clustering. We also present a new Arabic lemmatization test set covering diverse genres, standardized alongside existing datasets. We evaluate character-level sequence-to-sequence models, which perform competitively and offer complementary value, but are limited to lemma prediction (not LPG) and prone to hallucinating implausible forms. Our results show that classification and clustering yield more robust, interpretable outputs, setting new benchmarks for Arabic lemmatization.

1 Introduction

Lemmatization is the process of mapping a word to a base form that abstracts away from its inflectional variants. Lemmatization has played an important enabling technology role in many NLP applications, including machine translation (Conforti et al., 2018), information retrieval (Semmar et al., 2006), parsing (Seddah et al., 2010), text classification (Abdelrahman et al., 2021) and summarization (El-Shishtawy and El-Ghannam, 2014). Despite the shift toward large language models, lemmatization remains essential for tasks involving morphologically rich languages and requiring interpretability, such as readability assessment (Al Khalil et al., 2018; Liberato et al., 2024) or automated error detection (Belkebir and Habash, 2021).

Lemmatization is especially challenging in morphologically rich languages like Arabic due to complex morphology and optional diacritics. Table 1 presents multiple out-of-context analyses of a single word, varying in diacritization, lemma, POS, and English gloss (as a proxy for sense).

Stem		Lemma		POS	Gloss
عَقَد	*ςaqad*	عَقَد	*ςaqad*	verb	hold
عُقِد	*ςuqid*				be held
عَقَّد	*ςaq~ad*	عَقَّد	*ςaq~ad*	verb	complicate
عَقْد	*ςaq.d*	عَقْد	*ςaq.d*	noun	holding contract decade
عِقْد	*ςiq.d*	عِقْد	*ςiq.d*	noun	necklace
عُقَد	*ςuqad*	عُقْدَة	*ςuq.daħ*	noun	complexes

Table 1: Eight possible Lemma-POS-Gloss analyses for the Arabic word عقد *ςqd*. Transliteration is in the HSB scheme (Habash et al., 2007).

Previous lemmatization approaches rely on morphological analyzers and ranking models (Roth et al., 2008), sequence-to-sequence (seq2seq) generation (Bergmanis and Goldwater, 2018a; Zalmout and Habash, 2020), or edit-based tagging (Gesmundo and Samardžić, 2012; Kondratyuk et al., 2018a). However, these methods often focus only on the lemma form, lack generalization across domains, and rely on narrow lexical resources or genre-specific training data. In this work, we propose a broader and more interpretable framing of lemmatization as classification into a rich Lemma-POS-Gloss (LPG) tagset. Our contributions are:[1]

First, we introduce two novel approaches that classify into LPG labels: (a) leveraging machine translation of source sentences and dictionary glosses, and (b) using LPG semantic clustering.

Second, we present a new multi-genre Arabic lemmatization test set, covering underexplored domains such as novels and children's stories.

Our experiments demonstrate that LPG-based classification and clustering approaches outperform prior systems that resolve most morphosyntactic

[1] Code, models, and annotations: https://github.com/CAMeL-Lab/lemmatization-as-classification

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 30014–30029

Unique Avg	All	Top	Ambig↓	Recall
Analyses	15.5	1.3	91.3%	
LPG	2.7	1.3	52.8%	96.2%
LP	2.5	1.2	52.6%	98.8%
L	2.0	1.2	42.4%	99.6%

Table 2: Avg # of unique entries of CAMeL Tools analyzer and disambiguator on the ATB Dev set in terms of full morphological analyses, Lemmas (**L**), Part-of-Speech (**P**) and Gloss (**G**) combinations. **All** refers to all returned unique values per word, and **Top** refers to all remaining values after filtering with the POS Tagger. **Ambig↓** shows the effect of the POS Tagger. **Recall** shows the maximal potential accuracy for each representation combination.

ambiguity (Inoue et al., 2022), offering superior accuracy and robustness. We also evaluate character-level seq2seq models, which perform competitively and provide complementary benefits, but are limited to lemma-only (not LPG) prediction and often hallucinate implausible forms. Hybrid models that combine seq2seq and classification techniques further boost performance.

The paper is organized as follows: §2 covers linguistic background, §4.2 reviews related work, §4 describes the dataset, §5 outlines our methods, and §6 presents the evaluation results.

2 Linguistic Background

Arabic is morphologically and orthographically rich, with optional diacritics and multiple word forms contributing to ambiguity in both meaning and structure. Previous research has focused on lemma alone (L) or lemma with POS (LP), but none have examined the more complex Lemma, POS, and Gloss (LPG). This study aims to fill this gap while evaluating simpler variations for completeness.

We use the CAMeL Tools analyzer-and-disambiguator system as our baseline (Inoue et al., 2022; Obeid et al., 2020), which returns a set of ranked morphological analyses per word, including gender, number, clitics, POS, and 37 other features. While this helps resolve many morphosyntactic ambiguities, it does not fully disambiguate the lemma or sense, which are often given the same rank. For instance, for بعقدها *bςqdhA*, the model correctly rules out a verbal interpretation, but ambiguity remains among nominal readings such as 'in her contract,' 'necklace,' or 'complexes' (see Table 1).

As shown in Table 2, we define ambiguity as the average number of analyses per word and measure its reduction as the relative decrease across processing stages. The analyzer initially produced an average of 15 analyses per word on the development set. Restricting to the top-ranked disambiguator output reduced this to 1.3, achieving a 91.3% reduction through morphosyntactic feature tagging. For LPG selection, ambiguity decreases from 2.7186 to 1.2831, yielding a 52.8% reduction, though capped at 96.2% recall. LPG also starts with a larger ambiguity space than LP 2.7 vs. 2.5 on average, representing a 108% relative ambiguity, which contributes to the greater difficulty and more pronounced impact on recall in the LPG setting.

3 Related Work

3.1 Lemmatization Resources

In Arabic lemmatization, morphological dictionaries and analyzers serve as the primary resources for nearly all previous works in this task (Maamouri et al., 2010; Boudchiche et al., 2017; Taji et al., 2018; Jarrar et al., 2024). These analyzers or dictionaries extract the lemmas in an out-of-context manner based on morphosyntactic features. While they provide a strong foundation for lemmatization, their reliance on predefined linguistic rules limits adaptability to contextual variations. In this paper, we make use of the CALIMA-S31 analyzer (Taji et al., 2018), which is a rich Arabic morphological analyzer. It offers detailed form-based and functional morphological features, tokenization, lexical rationality, and more, and it also extends SAMA31 (Maamouri et al., 2010), and is used inside CAMeL Tools(Obeid et al., 2020).

Several benchmark datasets exist for Arabic lemmatization, including the Penn Arabic Treebank (Maamouri et al., 2004), ZAEBUC (Habash and Palfreyman, 2022), Wiki News (Mubarak, 2018), Salma (Jarrar et al., 2024), Quran (Dukes and Habash, 2010), and NEMLAR (Yaseen et al., 2006). However, most are heavily skewed toward the news genre, limiting their applicability to diverse linguistic contexts. In addition, inconsistencies in lemma definitions and diacritic conventions complicate fair comparisons across systems (Elgamal et al., 2024). Table 3 highlights some of the differences using three example lemmas. To address this, we apply a synchronization method to standardize lemma and diacritic representations, enabling more consistent evaluation. We also introduce a new multi-genre benchmark dataset to expand coverage

ATB	ٱِسْتَطاع	*Äis.taTAς*	أَصْبَح	*ÂaS.baH*	هٰذا	*háðA*
BAREC	ٱِسْتَطَاع	*Äis.taTaAς*	أَصْبَح	*ÂaS.baH*	هَٰذَا	*haáðaA*
Nemlar	اسْتَطَاعَ	*As.taTaAςa*	أَصْبَحَ	*ÂaS.baHa*	ذَا	*ðaA*
Quran	ٱسْتَطَاعَ	*Äs.taTaAςa*	أَصْبَحَ	*ÂaS.baHa*	هَٰذَا	*haáðaA*
WikiNews	اِسْتَطَاعَ	*Ais.taTaAςa*	أَصْبَح	*ÂaS.baH*	هَذَا	*haðaA*
ZAEBUC	اِسْتَطاع	*Ais.taTAς*	أَصْبَح	*ÂaS.baH*	هَذا	*haðA*

Table 3: Examples highlighting differences in lemma representations across the data sets we synchronize: استطاع *AstTAς* 'be capable', أصبح *ÂSbH* 'become', and هذا *hðA* 'this'.

beyond news and support a more comprehensive assessment of lemmatization approaches. Most of the aforementioned datasets are included in our evaluation to ensure broad generalization.

3.2 Lemmatization Approaches

Lemmatization has been tackled through various paradigms. One common approach relies on morphological dictionaries, framing lemmatization as the selection of the correct lemma from a predefined lexicon (Jarrar et al., 2024; Mubarak, 2018; Jongejan and Dalianis, 2009; Ingason et al., 2008; Ingólfsdóttir et al., 2019). These methods use morphosyntactic features and heuristics, but often fail to generalize well in contextually diverse settings.

Other studies treat lemmatization as a language modeling task, predicting lemmas and associated features based on morphological analysis (Pasha et al., 2014; Obeid et al., 2022; Lagus and Klami, 2021). While these models leverage rich linguistic resources, they may struggle with out-of-vocabulary (OOV) forms and the complexities of highly inflected languages.

A third line of work frames lemmatization as a tagging task. Gesmundo and Samardžić (2012) model it as paradigm-based tagging, learning transformation rules over affixes instead of mapping directly to lemmas. This enables better generalization and efficient use of context. Müller et al. (2015) extends this idea with LEMMING, a joint log-linear model that simultaneously learns lemmatization and POS tagging, showing that the two tasks benefit from being learned together.

Sequence-to-sequence (seq2seq) models represent another family of approaches. Bergmanis and Goldwater (2018b) frame lemmatization as character-level translation, using an encoder-decoder architecture with context markers, based on the Nematus toolkit (Sennrich et al., 2017). Kondratyuk et al. (2018b) employ an autoregressive decoder with Luong attention and integrate POS and sentence context features. Other recent work further explores this direction using neural seq2seq models (Sahala, 2024).

This paper explores leveraging external language signals for disambiguation, reframing lemmatization as LPG (Lemma-POS-Gloss) classification rather than lemma-only prediction. We introduce a semantic cluster formulation to better handle LPG complexity.

4 Data

We report results using six existing datasets with lemmatization annotations: **ATB** (Maamouri et al., 2004), **NEMLAR** (Yaseen et al., 2006), **Quran** Corpus (Dukes and Habash, 2010), **Wiki News** (Mubarak, 2018), **ZAEBUC** (Habash and Palfreyman, 2022), and annotate a new dataset from the **BAREC** corpus (Elmadani et al., 2025; Habash et al., 2024).

4.1 Data Preparation and Synchronization

As mentioned earlier, previous research on Arabic lemmatization has shown inconsistencies in both task definition and lemma representation, particularly in diacritization (Table 3). Highlights some of the differences using three example lemmas. To address this, we align all datasets with CALIMA-S31 standards (Taji et al., 2018), which are based on the LDC Standard Arabic Morphological Analyzer (SAMA3.1) (Maamouri et al., 2010). The process involves ranking and selecting the closest LPG set for each word in a given dataset after applying normalization, and computing a synchronization score to determine the best-matching reference.

Rationale for CALIMA-S31 Alignment CALIMA-S31 follows the same lemma annotation and diacritization rules as the LDC. Since LDC's data (Arabic Treebank) constitutes a major linguistic resource, aligning with CALIMA-S31 ensures consistency across datasets. Also, the CALIMA-S31 morphological database is supported by the Camel Tools toolkit for Arabic NLP (Obeid et al., 2020).

Data LPG Synchronization Pipeline For each word, we retrieve all possible LPG sets from CALIMA-S31 and rank them based on a synchronization score. The LPG set with the highest score is selected as the gold reference. If multiple candidates achieve the same highest score, a backoff

Dataset	All Tokens	News	Evaluatable
ATB Train	503,015	100.0%	99.1%
ATB Dev	63,137	100.0%	99.2%
ATB Test	63,172	100.0%	99.0%
BAREC	98,676	18.5%	96.9%
NEMLAR	480,417	52.6%	98.4%
Quran	77,429	0.0%	100.0%
WikiNews	18,300	100.0%	100.0%
ZAEBUC	34,235	0.0%	100.0%
Total	1,338,381	68.6%	98.8%

Table 4: Dataset statistics: total token count, proportion from news text, and proportion with a gold lemma reference.

strategy resolves the ambiguity. To ensure consistency between the gold reference and CALIMA-S31 outputs, we apply several normalization steps addressing diacritic and orthographic variations. A detailed list is provided in Appendix C. Our normalization decisions aimed to standardize lemma representations across all resources without deletion or ambiguity. This was challenging due to variant forms, and we followed the guidelines of (Elgamal et al., 2024).

Calculation of Synchronization Score After retrieving LPG sets and applying normalization, we compute a synchronization score across each three LPG dimensions to determine the best-matching reference. All scores are normalized to fall within a range of 0 to 1. The score computation depends on the available data dimensions, i.e., LPG, LP, or L, as well as on the presence of an actual gold reference in the original data. The final choice is based on the highest synchronization score. A detailed explanation of score computation is in Appendix C.

As shown in Table 4, following the synchronization stage, each dataset is either fully or partially evaluatable. In some cases, portions remain non-evaluatable due to missing gold references in the original source, preventing complete alignment. This distinction ensures that only consistently annotated data is used for evaluation, supporting fair and reliable comparisons across datasets.

To ensure the effectiveness of the synchronization process, we conducted a manual error analysis by selecting 100 records from each of the seven datasets (which will be discussed in detail later). The results revealed only six errors across the 700 records, yielding an overall error rate of 0.86%.

Analyzer	OOV Words	OOV Rate
CALIMA-S31	1,398	1.40%
CAMeL Morph MSA	824	0.83%
Both Analyzers	779	0.79%

Table 5: OOV word counts and percentages in the new dataset across different analyzers

4.2 Datasets

We evaluate our approaches across the listed datasets. As shown in Table 4, news data accounts for nearly twice as much as all other genres combined, and our baseline disambiguator is trained exclusively on news text (ATB Train).

We also introduce a new benchmark dataset based on a portion of the publicly available BAREC Corpus (Elmadani et al., 2025; Habash et al., 2024). The **BAREC Lemmatization Dataset** comprises diverse genres like 1001 Nights, Poetry, Novels, Emarati Curriculum, ChatGPT, Subtitles, Sahih al-Bukhari, and others (See Table 10 in Appendix A). We annotated this dataset following the standard lemmatization guidelines used in (Maamouri et al., 2010), and included the lemma, POS, and gloss for each word using CAMeL Morph MSA (Khairallah et al., 2024), an open-source morphological database with very high coverage that goes beyond CALIMA-S31. The annotation was completed by one Arabic native speaker with extensive experience in Arabic annotation.

Table 5 presents a comparison of out-of-vocabulary words identified by CAMeL Morph MSA, CALIMA-S31, and those shared by both analyzers. CAMeL Morph MSA exhibits a lower OOV rate, though a notable overlap remains across both systems.

5 Approach

We investigate a range of approaches with varying reliance on existing lemmatization resources, primarily morphological analyzers and annotated corpora.[2] Table 6 summarizes the approaches and techniques explored in this study.

Our main classification approaches start from a set of LPG candidates per word produced by a morphological analyzer, either unranked (All) or ranked by a POS tagger (Top). A classifier selects

[2]While one can distinguish between out-of-context analyzers and in-context annotated corpora as different types of artifacts, we note that most annotated datasets depend on analyzer lexicons to support the manual annotation process.

Methodology	Technique	Required Resources
Sequence to Sequence	**S2S**	Character-level Transformer Generation Model + Annotated Corpus
Random Selection	**Rand**	Morphological Analyzer + Deterministic Randomization
Probabilistic Selection	**LogP**	Morphological Analyzer + Annotated Corpus
Disambiguator	**Tagger**	Morphological Analyzer + Annotated Corpus + Tagger
Gloss Cosine Similarity	**SimG**	Morphological Analyzer + Machine Translation + Sim Align + Sent Similarity LM
Classification	**LexC**	Classifier + Annotated Corpus
	LexC+Tagger	Morphological Analyzer + Annotated Corpus + Classifier
Clustering	**Clust**	Morphological Analyzer + Annotated Corpus + Clustering Model

Table 6: A summary of all the approaches and techniques used in this study, along with the required resources needed for implementation in any language.

among these candidates. We also evaluate classification without an analyzer, using only the annotated corpus. We explore different classifier types that leverage various input features and model architectures. Additionally, we test a seq2seq model that directly predicts the lemma from the input word and its context, without relying on analyzer-generated options. Finally, we investigate hybrid models that combine these techniques.

We discuss the various approaches next.

Sequence to Sequence model (S2S) We trained a sequence-to-sequence model from scratch using the ATB training data. The input to the model consists of the target word along with a context window of two words before and two words after, while the output is the corresponding lemma for the target word. This setup enables the model to learn contextual patterns that inform lemma generation without relying on predefined candidate sets. Details are in Section 6.

Random Selection (Rand) As a simple baseline, we select an LPG candidate randomly using a deterministic method: the word's index modulo the number of candidates.

Probabilistic Selection (LogP) In this approach, the system retrieves all possible LPG candidates and ranks them based solely on the log probability of the lemma and POS combination. The top-ranked candidate is selected as the final output.

Disambiguator (Tagger) This method extends probabilistic selection by first ranking LPG candidates using POS tagger scores, then sorting by lemma and POS log probabilities from the annotated corpus. The top candidate is chosen, serving as our main probabilistic baseline.

Gloss Cosine Similarity (SimG) In this approach, re-ranking is based on the cosine similarity between each gloss in the LPG set and its aligned English counterpart from the translated sentence. The translation is generated using the Google Translate API, and word alignment is performed using the SimAlign RoBERTa model with the 'mwmf' alignment strategy (Jalili Sabet et al., 2020). Both the gloss and the aligned English word are embedded using the gte-Base English language model (Li et al., 2023), and similarity is computed using cosine similarity. If alignment fails for a given Arabic word, the similarity is instead computed between the gloss and the entire translated sentence.

Classification (LexC) In this approach, lemmatization is framed as a classification task, where each unique LPG is treated as a distinct class, resulting in approximately 18,000 target classes that the model has encountered during training. To reduce noise, digits and punctuation are grouped into a single class, while words lacking a gold reference are assigned to a separate "unknown" class. A BERT model is fine-tuned in two stages. In the first stage, the model is trained using the input word along with its context, with the corresponding LPG class as the target label. In the second stage, instances without a gold label are reassigned to the most probable class based on predictions from the first model. The model is then fine-tuned again using these updated labels to further improve accuracy. The final fine-tuned model is used to select the most suitable LPG from the candidate set or to fall back on alternative selection strategies when needed.

The final fine-tuned model is utilized in two different ways: (1) directly using its prediction as the LPG (LexC), (2) checking if the predicted LPG exists in the primary LPG set from the analyzer; if it does, selecting it; otherwise, applying a fallback strategy reverting to the primary log probability-based ranking (LexC+LogP).

Clustering (Clust) In this approach, we redefine lemmatization as a clustering task, which is later transformed into a classification problem. Each unique LPG is grouped into a cluster with semantically similar entries (e.g., countries forming one cluster and cities another). Clusters are formed using a fine-tuned classification model combined with a clustering technique for known LPGs, with the number of clusters determined based on a custom evaluation metric. For unknown LPGs in the morphological database that have not yet been assigned a cluster, gloss-based cosine similarity is applied to identify the closest existing cluster, to which they are then assigned. The motivation behind this method is to reduce the search space for identifying the correct LPG from the LexC method by narrowing the candidate set to a smaller, semantically organized group. In total, we arrived at 2,000 clusters that collectively represent the entire LPG space of the analyzer. A sample of these clusters is provided in Appendix D.

Once the clusters are established, a classification model is fine-tuned to predict the cluster containing the correct LPG from the given primary set. If any LPGs in the primary set belong to the predicted cluster, they are extracted and re-ranked based on POS-LEX log probability, with the top-ranked option selected. If no LPGs from the primary set match the predicted cluster, the system falls back to the primary backoff technique.

The clustering process relies on a fine-tuned model to generate contextual embeddings for each word in the training set. Words that share the same LPG are assigned the same averaged embedding. These embeddings are then clustered using the K-Means algorithm. To determine the optimal number of clusters, we introduce a metric called the Cluster Compactness Ratio (CCR). This ratio is calculated as the average number of ambiguous lemmas that share the same cluster per word, divided by the total number of ambiguous lemmas. Intuitively, the goal is to minimize this value, since an ideal clustering would assign each ambiguous lemma to its own distinct cluster. By encouraging separation between competing lemmas, the CCR helps guide the selection of a cluster count that reduces ambiguity and results in tighter, semantically coherent groupings. This favors smaller, well-defined clusters over broader ones, improving the reliability of the final lemma selection process. Once the clusters are formed, a clustering model is fine-tuned specifically on the cluster labels, as previously discussed, to further refine the selection process.

Table 6 presents a summary of all the approaches and techniques used in this study, along with the required resources needed for implementation in any language. Each method varies in computational complexity based on its dependencies. As highlighted, the most computationally expensive techniques are SimG, primarily due to their reliance on external machine translation systems, such as Google API, and alignment models that are pretrained neural models rather than statistical ones. While these methods improve disambiguation via contextual similarity, their practicality is limited by computational overhead. In contrast, classification and clustering approaches, though requiring fine-tuned models, are generally more efficient, scalable, and easier to optimize.

6 Evaluation

6.1 Experimental Setups

Data The data used for training the unigram log probability model and fine-tuning the classification and clustering models was derived from the ATB123 Train set, following the same splits outlined in the literature (Diab et al., 2013; Khalifa et al., 2020; Inoue et al., 2022) or provided by the data set creators. This ensures consistency with prior work and enables direct comparison of results across different methodologies.

Metrics Results report accuracy over L, LP, or LPG matches on evaluatable data, counting all tokens.

Building the Sequence to Sequence Model We trained a sequence-to-sequence model for Arabic lemmatization from scratch, using a 6-layer encoder-decoder architecture with 6 attention heads, a hidden size of 512, and a feed-forward dimension of 2048. Dropout was set to 0.2 across all components, and input sequences were capped at 64 tokens to match the context window size. The model was trained without caching and initialized with the padding token for decoding.

Training was conducted on three parallel NVIDIA A100 GPUs and completed in approximately 5 hours. We used Hugging Face's `Seq2SeqTrainer` with a learning rate of 5×10^{-5}, batch sizes of 64 (train) and 32 (eval), and 100 epochs. Gradient checkpointing and FP16 precision were enabled to optimize memory and speed.

	Technique	Corpus	Tagger	Analyzer	Classifier	Generator	Select	L	LP	LPG
(a)	**S2S**	L	-	-	-	S2S	-	95.0	-	-
(b)	**LexC**	LPG	-	-	LexC	-	-	89.5	88.5	85.6
	LexC+S2S	LPG	-	-	LexC	S2S	-	95.0	90.0	74.9
(c)	**All+Rand**	-	-	AllSet	-	-	Rand	72.9	64.6	59.4
	All+SimG	-	-	AllSet	-	-	SimG	91.7	87.0	83.2
(d)	**All+LogP**	LP	-	AllSet	-	-	LogP	93.7	91.4	88.2
	All+S2S+LogP	LP	-	AllSet	-	S2S	LogP	97.4	95.0	91.6
(e)	**Top+Rand**	P	POS	TopSet	-	-	Rand	93.0	92.3	87.1
	Top+SimG	P	POS	TopSet	-	-	SimG	98.1	97.3	94.3
(f)	**Top+LogP**	LP	POS	TopSet	-	-	LogP	98.2	97.4	94.4
	Top+S2S+LogP	LP	POS	TopSet	-	S2S	LogP	98.7	97.9	94.9
(g)	**Top+LexC+LogP**	LPG	POS	TopSet	LexC	-	LogP	98.8	**98.1**	**95.6**
	Top+LexC+S2S+LogP	LPG	POS	TopSet	LexC	S2S	LogP	**98.9**	**98.1**	**95.6**
(h)	**Top+Clust+LogP**	LPG	POS	TopSet	Clust	-	LogP	98.8	**98.1**	95.4
	Top+Clust+S2S+LogP	LPG	POS	TopSet	Clust	S2S	LogP	**98.9**	**98.1**	95.4

Table 7: Comparison of techniques across different configurations on the ATB Dev set. The table summarizes the components used in each setup, including the corpus type, tagger, analyzer, classifier, generator, and tiebreaking method.

The best model was selected based on validation accuracy evaluated at the end of each epoch.

Fine-Tuning for Classification and Clustering The CAMeL BERT msa_pos_MSA model (Inoue et al., 2021) is fine-tuned for both classification and clustering tasks. Training is performed over 10 epochs with a learning rate of 2×10^{-5}, a batch size of 16, and a maximum sequence length of 512. Three fine-tuned models were trained on an A100 GPU, with an estimated training time of 1 hour per model.

Disambiguator & Morphological Analyzer DB All experiments use the CAMeL-unfactored BERT disambiguator model as the baseline competitor (Inoue et al., 2022), with CALIMA-S31 (Taji et al., 2018) as the morphological analyzer DB and NOAN_PROP as the backoff technique, as implemented in CAMEL Tools (Obeid et al., 2020).[3]

We first evaluated all our approaches on the ATB123 dev set, as presented in Table 7. These approaches were assessed using three different evaluation granularities, as previously mentioned. Our experiments on the dev set were conducted in a variety of configurations, each representing a distinct combination of the proposed techniques. For each configuration, we considered multiple factors: whether the technique operates independently or depends on prior annotation, whether it relies on an external tagger, whether it incorporates the morphological analyzer, whether it has access to the full set of LPG candidates or only the top-ranked option, and whether it integrates outputs from the classification or clustering models. Whether the technique includes a generator (e.g., seq2seq model) and how tie-breaking is handled when multiple candidates remain.

Each technique or combination of techniques is treated as a sequential pipeline. For example, the setup "Top+Clust+S2S+LogP" follows a sequential process: retrieve the top-ranked LPG set, filter by predicted cluster, match the seq2seq-predicted lemma, and finally select the candidate with the highest log probability. This modular evaluation framework allows us to compare the contribution of each component under controlled conditions.

6.2 Results

Results in Table 7 highlight several insights about the performance of different lemmatization strategies. In group (a), the seq2seq model trained independently of the analyzer achieves strong results, outperforming the LexC classifier when used on its own in group (b), and when combined with seq2seq, this approach not only leverages the advantage of a generative model that is unconstrained by a predefined candidate set, but also incorporates the benefits of LexC, enabling the inclusion of POS tags and glosses for richer linguistic representation.

Group (c) focuses on setups using only the analyzer. The random selection method (All+Rand)

[3]CamelTools v1.5.5: Bert-Disambig+calima-msa-s31 db.

Dataset	ATB Test			BAREC			NEMLAR	Quran		WikiNews		ZAEBUC		
Tag	L	LP	LPG	L	LP	LPG	L	L	LP	L	LP	L	LP	LPG
S2S	95.0	-	-	87.0	-	-	83.6	65.7	-	90.5	-	92.5	-	-
LexC+S2S	95.0	90.4	75.0	87.0	78.0	64.2	83.6	65.7	61.6	90.5	86.9	92.5	90.5	77.0
All+Rand	73.1	64.7	59.8	69.9	62.7	57.7	62.4	55.3	46.7	68.6	61.1	67.0	61.6	57.2
Top+Rand	92.9	92.2	87.3	90.2	89.1	83.8	84.0	77.8	75.7	89.1	87.8	90.8	89.5	84.3
Top+LogP	98.0	97.3	94.6	96.4	95.3	92.1	89.6	83.2	81.0	94.4	93.1	96.2	94.8	91.0
Top+S2S+LogP	98.6	97.9	95.2	96.6	95.5	92.4	90.2	83.3	81.1	94.9	93.5	97.0	95.6	91.7
Top+LexC+LogP	98.7	98.0	95.9	**97.1**	**96.0**	**92.7**	**90.5**	**84.5**	**82.3**	95.0	93.7	97.3	95.9	92.1
Top+LexC+S2S+LogP	98.7	**98.1**	**96.0**	97.0	95.9	92.6	**90.5**	**84.5**	**82.3**	95.1	93.7	97.3	96.0	92.1
Top+Clust+LogP	98.7	98.0	95.5	**97.1**	**96.0**	92.6	**90.5**	**84.5**	**82.3**	**95.2**	**93.8**	**97.5**	**96.1**	**92.2**
Top+Clust+S2S+LogP	**98.8**	**98.1**	95.7	97.0	95.9	92.6	**90.5**	84.1	81.9	95.1	93.7	97.3	95.9	92.0

Table 8: Performance of different systems evaluated on multiple **test sets** across varying tagset granularities.

performs poorly, but adding gloss-based similarity (All+SimG) significantly improves results, demonstrating the usefulness of semantic signals in the absence of other models.

In group (d), introducing lemma and POS information (LP) through log probability ranking improves performance, and adding the seq2seq model further boosts accuracy by helping narrow down the correct lemma more precisely.

Groups (e) and (f) evaluate scenarios with access to POS tags and only the top-ranked candidates from the tagger. These represent practical, efficient setups. The "Top+LogP" method provides a strong baseline, and using the seq2seq model as a filter (Top+S2S+LogP) improves it even further.

Finally, groups (g) and (h) incorporate richer supervision through LexC classification or LPG clustering. Both yield better results and outperform the baseline, with the LexC approach achieving higher performance, particularly on the LPG set. Adding the seq2seq model to both techniques provides a small but consistent improvement, further enhancing these already strong configurations

In Table 8, the results across the various test sets are largely consistent with the patterns observed on the ATB123 dev set, reinforcing the generalizability and robustness of our proposed methods. Notably, the clustering-based approach demonstrates superior performance across most of the datasets for the lemma granularity (Top+Clust+LogP). This highlights the strength of semantically informed clustering in capturing lexical variation and guiding lemma selection, even in diverse and unseen domains. However, for other granularities, the two classification and clustering methods show competitive performance against each other.

We measure statistical significance using the McNemar Test (McNemar, 1947), applied at the highest available granularity for each test set (Table 8). All improvements of Top+Clust+LogP and Top+LexC+S2S+LogP over Top+LogP are statistically significant ($p < 0.05$). Furthermore, all pairwise differences between (Top+Clust+LogP and Top+LexC+LogP) and between (Top+Clust+S2S+LogP and Top+LexC+S2S+LogP) are statistically significant ($p < 0.05$), with the exception of the Quran dataset in the comparison between (Top+Clust+LogP and Top+LexC+LogP) and the BAREC dataset in both comparisons.

This performance difference may be attributed to the fact that the clustering technique considers and leverages information from the entire 49K LPG entries in the CALIMA-S31 database, whereas the classification-based approach is limited to approximately 18K unique classes. By incorporating a broader range of lexical knowledge, clustering may offer a more comprehensive representation, contributing to its advantage in certain datasets.

6.3 Error Analysis

We conducted a manual error analysis to better understand the failure cases of our best-performing system (**Top+Clust+LogP**, henceforth **BEST**) compared to the character-level sequence-to-sequence model (**S2S**) on the ATB dev set. Out of 62,609 evaluatable entries, S2S made 3,108 errors, BEST made 708, with 631 errors overlapping (20% of S2S, 89% of BEST).

To gain insight, we randomly sampled 100 errors from the **S2S only** set, 100 from the **S2S+BEST** overlap, and included all 77 **BEST only** errors. Below, we report on the 200 S2S and the 177 BEST errors (Table 9).

Error Type	S2S	BEST
Hallucination	40.0%	0.0%
Plausible	26.5%	52.5%
Diacritization/Hamzation	33.5%	47.5%

Table 9: S2S and BEST system error type distributions.

We categorized the errors into three types:

(a) Hallucination: The predicted lemma is not morphologically plausible, e.g., the word ترحيباً *trHybAã* (reference lemma تَرْحِيب *tar.Hiyb* '*welcome*') was lemmatized by S2S as تَحَرِّي *taHar∼iy 'investigation'*.

(b) Plausible: The predicted lemma is morphologically valid but differs noticeably from the reference, the word لزهرة *lzhrħ* (reference lemma زَهْر *zah.r* 'flower') is lemmatized as زُهْرَة *zuh.raħ* 'Venus'.

(c) Diacritization/Hamzation: The predicted lemma differs from the reference primarily by diacritics or hamza placement, e.g., the word وتحولها *wtHwlhA* (reference lemma تَحَوُّل *taHaw∼ul* 'change [noun]') is lemmatized as تَحَوَّل *taHaw∼al* 'change [verb]'.

S2S often hallucinated implausible lemmas (40%), while BEST showed no hallucinations and mostly subtle diacritic or variant errors, indicating classification methods produce more morphologically consistent lemmas for Arabic.

7 Conclusion and Future Work

We introduced new lemmatization methods by framing the task as classification and clustering in the Lemma-POS-Gloss (LPG) space. Evaluated across multiple Arabic datasets (with synchronized benchmarks for consistency) and compared to character-level seq2seq models, our approaches showed strong cross-genre generalization and added-value hybridization. Our models also avoided the hallucination issues seen in seq2seq outputs. Significance testing confirmed that all performance gains were statistically meaningful.

We will release all annotations, synchronizations, and code to support future work. Going forward, we aim to expand training data, improve analyzer recall with broader LPG candidate generation, retrain the models on more diverse corpora, and explore seq2seq as a fallback for OOV terms to further boost robustness.

Limitations

The classification-based model is constrained by a predefined set of approximately 18,000 LPG classes, while the clustering-based model operates over 2000 LPG clusters. Both approaches face challenges with out-of-vocabulary (OOV) lemmas, as fallback strategies may fail to select the optimal lemma even when it is present in the known sets. Moreover, relying solely on the top-ranked LPG candidate from the disambiguator can reduce recall by eliminating potentially correct alternatives. As for the sequence-to-sequence (S2S) model, error analysis revealed that it occasionally hallucinates lemmas not grounded in the input, especially in ambiguous contexts. Its performance may further improve if trained on datasets spanning a broader range of genres, allowing it to generalize better to lexical variations and domain-specific usage.

Ethics Statement

All data used in the corpus collection and curation process are sourced responsibly and legally. The annotation process is conducted with transparency and fairness. The Arabic native speaker annotator who helped with the BAREC lemmatization Dataset was paid fair wages for their contribution. We acknowledge that enabling technologies such as lemmatization can be used with malicious intent to profile people based on their lexical choices or be used to build malicious software; this is not our intention, and we discourage it. We used AI writing assistance within the scope of "Assistance purely with the language of the paper" described in the ACL Policy on Publication Ethics.

Acknowledgments

We acknowledge the support of the High Performance Computing Center at New York University Abu Dhabi.

References

Eman Abdelrahman, Fatimah Alotaibi, Edward A Fox, and Osman Balci. 2021. Otrouha: A corpus of arabic etds and a framework for automatic subject classification. *The Journal of Electronic Theses and Dissertations*, 1(1):6.

Muhamed Al Khalil, Hind Saddiki, Nizar Habash, and Latifa Alfalasi. 2018. A Leveled Reading Corpus of Modern Standard Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Riadh Belkebir and Nizar Habash. 2021. Automatic error type annotation for Arabic. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 596–606, Online. Association for Computational Linguistics.

Toms Bergmanis and Sharon Goldwater. 2018a. Context sensitive neural lemmatization with lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1391–1400.

Toms Bergmanis and Sharon Goldwater. 2018b. Context sensitive neural lemmatization with lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1391–1400.

Mohamed Boudchiche, Azzeddine Mazroui, Mohamed Ould Abdallahi Ould Bebah, Abdelhak Lakhouaja, and Abderrahim Boudlal. 2017. Alkhalil morpho sys 2: A robust arabic morpho-syntactic analyzer. *Journal of King Saud University-Computer and Information Sciences*, 29(2):141–146.

Costanza Conforti, Matthias Huck, and Alexander Fraser. 2018. Neural morphological tagging of lemma sequences for machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 39–53.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic treebanks and associated corpora: Data divisions manual. *arXiv preprint arXiv:1309.5652*.

Kais Dukes and Nizar Habash. 2010. Morphological Annotation of Quranic Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Valetta, Malta.

Tarek El-Shishtawy and Fatma El-Ghannam. 2014. A lemma based evaluator for semitic language text summarization systems. *arXiv preprint arXiv:1403.5596*.

Salman Elgamal, Ossama Obeid, Mhd Kabbani, Go Inoue, and Nizar Habash. 2024. Arabic diacritics in the wild: Exploiting opportunities for improved diacritization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14815–14829, Bangkok, Thailand. Association for Computational Linguistics.

Khalid N. Elmadani, Nizar Habash, and Hanada Taha-Thomure. 2025. 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.

Andrea Gesmundo and Tanja Samardžić. 2012. Lemmatisation as a tagging task. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 368–372, Jeju Island, Korea. Association for Computational Linguistics.

Nizar Habash and David Palfreyman. 2022. ZAEBUC: An annotated Arabic-English bilingual writer corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 79–88, Marseille, France. European Language Resources Association.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, pages 15–22. Springer, Netherlands.

Nizar Habash, Hanada Taha-Thomure, Khalid N Elmadani, Zeina Zeino, and Abdallah Abushmaes. 2024. Guidelines for fine-grained sentence-level arabic readability annotation. *arXiv preprint arXiv:2410.08674*.

Anton Karl Ingason, Sigrún Helgadóttir, Hrafn Loftsson, and Eiríkur Rögnvaldsson. 2008. A mixed method lemmatization algorithm using a hierarchy of linguistic identities (holi). In *Advances in natural language processing: 6th international conference, GoTAL 2008 Gothenburg, Sweden, August 25-27, 2008 Proceedings*, pages 205–216. Springer.

Svanhvít Lilja Ingólfsdóttir, Hrafn Loftsson, Jón Friðrik Daðason, and Kristín Bjarnadóttir. 2019. Nefnir: A high accuracy lemmatizer for Icelandic. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 310–315, Turku, Finland. Linköping University Electronic Press.

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, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Go Inoue, Salam Khalifa, and Nizar Habash. 2022. Morphosyntactic tagging with pre-trained language models for Arabic and its dialects. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1708–1719, Dublin, Ireland. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Mustafa Jarrar, Diyam Akra, and Tymaa Hammouda. 2024. Alma: Fast lemmatizer and pos tagger for arabic. *Procedia Computer Science*, 244:378–387.

Bart Jongejan and Hercules Dalianis. 2009. Automatic training of lemmatization rules that handle morphological changes in pre-, in-and suffixes alike. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 145–153.

Christian Khairallah, Salam Khalifa, Reham Marzouk, Mayar Nassar, and Nizar Habash. 2024. Camel morph msa: A large-scale open-source morphological analyzer for modern standard arabic. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2683–2691.

Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2020. Morphological analysis and disambiguation for gulf arabic: The interplay between resources and methods. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3895–3904.

Daniel Kondratyuk, Tomáš Gavenčiak, Milan Straka, and Jan Hajič. 2018a. Lemmatag: Jointly tagging and lemmatizing for morphologically rich languages with brnns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4921–4928.

Daniel Kondratyuk, Tomáš Gavenčiak, Milan Straka, and Jan Hajič. 2018b. LemmaTag: Jointly tagging and lemmatizing for morphologically rich languages with BRNNs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4921–4928, Brussels, Belgium. Association for Computational Linguistics.

Jarkko Lagus and Arto Klami. 2021. Learning to lemmatize in the word representation space. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 249–258.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

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.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Mohamed Maamouri, Dave Graff, Basma Bouziri, Sondos Krouna, Ann Bies, and Seth Kulick. 2010. Ldc standard Arabic morphological analyzer (sama) version 3.1.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Hamdy Mubarak. 2018. Build fast and accurate lemmatization for Arabic. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. 2015. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon, Portugal. Association for Computational Linguistics.

Ossama Obeid, Go Inoue, and Nizar Habash. 2022. Camelira: An Arabic multi-dialect morphological disambiguator. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 319–326, Abu Dhabi, UAE. Association for Computational Linguistics.

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, Marseille, France. European Language Resources Association.

Arfath Pasha, Mohamed Al-Badrashiny, Ahmed El Kholy, Ramy Eskander, Mona Diab, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *In Proceedings of LREC*.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, Columbus, Ohio.

Aleksi Sahala. 2024. Neural lemmatization and pos-tagging models for coptic, demotic and earlier egyptian. In *Proceedings of the 1st Workshop on Machine Learning for Ancient Languages (ML4AL 2024)*, pages 87–97.

Djamé Seddah, Grzegorz Chrupała, Özlem Çetinoğlu, Josef van Genabith, and Marie Candito. 2010. Lemmatization and lexicalized statistical parsing of morphologically-rich languages: the case of French. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 85–93, Los Angeles, CA, USA. Association for Computational Linguistics.

Nasredine Semmar, Meriama Laib, and Christian Fluhr. 2006. A deep linguistic analysis for cross-language information retrieval. In *LREC*, pages 2507–2510.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Dima Taji, Salam Khalifa, Ossama Obeid, Fadhl Eryani, and Nizar Habash. 2018. An Arabic morphological analyzer and generator with copious features. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 140–150, Brussels, Belgium. Association for Computational Linguistics.

Mustafa Yaseen, Mohammed Attia, Bente Maegaard, Khalid Choukri, Niklas Paulsson, Salah Haamid, Steven Krauwer, Chomicha Bendahman, Hanne Fersøe, Mohsen A Rashwan, et al. 2006. Building annotated written and spoken arabic lrs in nemlar project. In *LREC*, pages 533–538. Citeseer.

Nasser Zalmout and Nizar Habash. 2020. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8297–8307, Online. Association for Computational Linguistics.

A BAREC and NEMLAR Distributions

Tables 10 and 11 illustrate the genre distribution within the BAREC and NEMLAR datasets, respectively. As shown in Table 10, BAREC covers a diverse range of genres, contributing to the increased complexity and challenge of processing this dataset. Similarly, Table 11 presents the distribution across various genres in NEMLAR, highlighting its wide coverage and relevance for evaluating lemmatization systems across different domains.

Category	**Words Count**
1001 Nights	4,607
ChatGPT	2,523
Emarati Curriculum	30,789
Hayy ibn Yaqzan Novel	1,038
Hindawi	10,450
Mama Makes Bread	416
My Language Enriches	1,843
Poetry and News	1,190
Quran	585
Sahih al-Bukhari	4,234
Sara (Al-Aqqad) Novel	1,165
Subtitles	3,374
Suleiman Al-Issa's Poetry	342
The Cat and the Eid Hat	246
The Mu'allaqat	1,526
The Notebook	2,327
UN	1,270
WikiNews	18,233
Wikipedia	12,518
Total	**98,676**

Table 10: The distribution of various genres within the BAREC dataset.

Category	**Words Count**
News	252,711
ArabicDictionaries	45,212
ArabicLiterature	29,701
Business	19,297
Interviews	47,218
IslamicTopics	28,877
Legal	26,922
PhrasesOfCommonWords	3,664
PoliticalDebate	26,815
Total	**480,417**

Table 11: The distribution of all genres within the NEMLAR dataset.

B License

In Table 12, we list the license of the data and tools used in this work. All of them are used under their intended use.

Data/tool	License
Arabic Treebank: Part 1 v 4.1 (LDC2010T13)	LDC User Agreement for Non-Members
Arabic Treebank: Part 2 v 3.1 (LDC2011T09)	LDC User Agreement for Non-Members
Arabic Treebank: Part 3 v 3.2 (LDC2010T08)	LDC User Agreement for Non-Members
BAREC (Elmadani et al., 2025)	Creative Commons Attribution-NonCommercial-ShareAlike 4.0
NEMLAR (Yaseen et al., 2006)	Non Commercial Use - ELRA END USER
Quran (Dukes and Habash, 2010)	GNU General Public License
WikiNews (Mubarak, 2018)	Creative Commons Attribution 4.0 License
ZAEBUC (Habash and Palfreyman, 2022)	Creative Commons Attribution-NonCommercial-ShareAlike 4.0
CAMeL Tools (Obeid et al., 2020)	MIT License
CAMeLBERT (Inoue et al., 2021)	MIT License

Table 12: License of the data and tools.

C Data Preparation & Synchronization

This section outlines the normalization procedures and scoring criteria used during the data synchronization stage. These steps ensure consistency between CALIMA-S31 outputs and the reference annotations.

Normalization Procedures The following normalization operations were applied in the exact order presented below to address diacritic inconsistencies, orthographic variations, and dataset-specific irregularities:

- **Alef Maqsura Normalization:** Convert Alef Maqsura following Kasra (ِى *iý*) into Yeh (ِي *iy*).
- **Shadda Order Correction:** Ensure Shadda always precedes any associated diacritic.
- **Alef Wasla Standardization:** Replace Alef Wasla followed by Kasra (ٱِ *Äi*) with a bare Alef (إ *Ai*).
- **Diacritic Removal in Long Vowel Spelling:** Remove diacritics in diacritic-letter sequences indicating long vowels: َا *aA* → ا *A*, ُو *uw* → و *w*, and ِي *iy* → ي *y*.
- **Dagger Alef Adjustment:** Replace Dagger Alef (ٰ *á*) and Fatha+Dagger Alef (َٰ *aá*) with Fatha (َ *a*).
- **Tanween Positioning:** Shift Tanween to the end of the word, e.g., ايضًا *AyDãA* → ايضاً *AyDAã*.
- **Final Letter Diacritics:** Remove all diacritics on the last letter except Shadda.
- **Sun Letter Shadda Removal:** Remove erroneous lemma-initial Sun Letter Shaddas, specifically in the Quran corpus (Dukes and Habash, 2010).
- **Alef Wasla Normalization:** Normalize Alef Wasla (ٱ *Ä*) to Alef (ا *A*).
- **Dataset-Specific Adjustments:** Certain datasets required additional handling for special cases. All synchronization procedures will be made publicly available.

Scoring Criteria The following criteria were used to compute synchronization scores and identify the best-matching LPG set for each token:

- **Lemma Score:** Assign a score of 1 if the predicted lemma matches the gold lemma. Otherwise, compute a penalty based on edit distance.
- **POS Score:** Assign 1 for a POS match and 0 for a mismatch.
- **Gloss Score:** Calculate the intersection between the gold gloss and each gloss in the CALIMA-S31 output.

D Examples of Clusters

Table 13 presents a representative sample of the lexical clusters generated automatically by the fine-tuned classification model. Each cluster groups together words that share similar semantic or functional properties, such as traits, foreign names, places, and vehicles.

Descriptive Attributes		Foreign Names		Locations		Veichles	
Word	**Gloss**	**Word**	**Gloss**	**Word**	**Gloss**	**Word**	**Gloss**
مؤثر	influential	الاباما	Alabama	استوديو	studio	اسطول	fleet
متازم	tense	اماندا	Amanda	اكر	farm;sharecrop	اوتوبيس	bus
مؤسف	regrettable	اند	And	انبار	warehouse	بابور	steamship
ماسوي	tragic	انديك	Indyk	ماب	resort	باص	bus
ماسوي	tragicness	انرون	Enron	اوي	shelter	باخر	steamship
مؤسي	saddening	انس	Anas	ابار	wells	بويخر	small_steamship
افين	stupid;dull	اوسكار	Oscar	بساتين	gardens	بارج	battleship
مؤلم	painful	ايان	Ian	متجر	store	بلم	anchovy
مبرح	agonizing	ايدي	Eddie	متاحف	museums	ابلام	sailing_barges
بشع	ugly	ايل	El	تياتروه	theaters	ابهر	dazzle
بطل	be_heroic	ايل	Il	جوابي	pools	زورق	boat
بليغ	eloquent	باتل	Patel;Batil	اجران	basins	جرم	barge
باهر	dazzling	باري	Paris	اجزائي	pharmacy	جلبوت	boat
مبهرج	gaudy;trashy	باولا	Paula	جفتلك	farm	حافل	bus
باهظ	oppressive	باولو	Paulo	جنائن	gardens	خافر	cruiser
متلف	damaging	بدرو	Pedro	حدائق	gardens	رفاس	steamboat
مثبط	discouraging	برادلي	Bradley	حقول	fields	رفل	train
مثمر	profitable	برفيز	Parvez	محال	places	مركب	ship;vessel

Table 13: Examples of words grouped into semantic clusters. Each word is paired with its English gloss.

E S2S Lex & Word INV/OOV Analysis

This analysis was conducted on the ATB Dev dataset to evaluate the model's accuracy when predicting both diacritized and undiacritized lemma forms. Since the model is trained as a character-level sequence-to-sequence system, we aimed to assess its sensitivity to surface diacritization (Table 14).

Case	Frequency	Predicted Words	Accuracy (%)
Diacritized			
Overall	62,609	59,495	95.0
(W-INV, L-INV)	57,963	56,878	98.1
(W-OOV, L-INV)	3,722	2,568	69.0
(W-INV, L-OOV)	48	1	2.1
(W-OOV, L-OOV)	876	49	5.6
Undiacritized			
Overall	62,609	60,208	96.2
(W-INV, L-INV)	57,963	57,188	98.7
(W-OOV, L-INV)	3,722	2,684	72.1
(W-INV, L-OOV)	48	17	35.4
(W-OOV, L-OOV)	876	319	36.4

Table 14: Prediction accuracy across diacritized and undiacritized inputs, broken down by in-vocabulary (INV) and out-of-vocabulary (OOV) word and lemma status.

F Lemma-Level Coverage Analysis Across Datasets

Table 15 presents a lemma-level analysis across all datasets used in the study. It categorizes each token based on whether its lemma is in-vocabulary (INV) or out-of-vocabulary (OOV) with respect to both the training set and the analyzer. Cases with no gold reference are marked as non-evaluatable.

Dataset	Train-INV Analyzer-INV	Train-INV Analyzer-OOV	Train-OOV Analyzer-INV	Train-OOV Analyzer-OOV	No Reference	Total
ATB_Train	498,430	0	0	0	4,585	503,015
All Tests						
ATB_Dev	61,740	0	869	0	528	63,137
ATB_Test	61,732	0	790	0	650	63,172
BAREC	91,941	0	3,185	501	3,049	98,676
NEMLAR	438,203	0	14,023	20,603	7,588	480,417
Quran	66,122	0	6,358	4,949	0	77,429
WikiNews	17,537	0	318	445	0	18,300
ZAEBUC	33,729	0	334	172	0	34,235
All Tests	**771,004**	**0**	**25,877**	**26,670**	**11,815**	**835,366**
Percentage	92.3%	0.0%	3.1%	3.2%	1.4%	

Table 15: Lemma-level analysis across all datasets used. This breakdown shows how many lemmas per dataset exist in the training data and/or analyzer (INV/OOV), and how many tokens have no gold reference, making them non-evaluatable.

G Analysis of Unseen Classes

To ensure that the model's performance is not merely a result of memorizing training data, we conducted an analysis of the ATB dev and test sets with a particular emphasis on unseen LPG classes defined as unique combinations of Lex, POS, and Gloss attributes. This analysis is intended to assess the model's ability to generalize to novel linguistic constructions rather than relying solely on previously encountered patterns. As detailed in Table 16, although a considerable proportion of LPG classes in both splits were also present in the training set (over 82%), a non-trivial number of previously unseen combinations remain. This supports the assertion that the task requires genuine generalization beyond memorization.

Split	Total Unique LPG Classes	Seen in Train	Seen in Train (%)
Dev Set	8,901	7,374	82.84%
Test Set	8,899	7,753	87.12%

Table 16: Overlap of LPG classes (Lex, POS, Gloss triplets) between the Train set and the Dev/Test sets.