Continual Multitask Active Learning for Multidialectal Arabic Morphological Tagging

Anonymous ACL submission

Abstract

Multidialectal Arabic morphological tagging is challenging due to rich morphology and strong dialectal variation. Building fully annotated dialect corpora is expensive, and scaling data does not always improve performance proportionally. Active learning reduces annotation cost by selecting the most informative samples, but is usually limited to single tasks such as POS tagging, or lemmatization. In this paper, we propose a continual multitask active learning framework for morphological tagging across Arabic dialects. Results show that an *alternating selection* strategy performs best. The model stabilizes at about 16,000 tokens across tasks, while prefix tagging stabilizes earlier ($\sim 8,000$ tokens) and lemmatization later ($\sim 22,000$ tokens). In continual learning, the *Uncertainty-Common* method reduces catastrophic forgetting, recovering $\sim 16\%$ POS tagging performance over random replay and cutting iteration time by ~ 2.5 minutes per iteration compared to full re-finetuning. Code, data, and models will be released as open source upon acceptance.

1 Introduction

Morphological tagging involves assigning morphological tags to each word in a sentence (Inoue et al., 2022; Darwish et al., 2017). These tags represent features such as POS, gender, number, aspect, stem, affixes, and lemma. It encompasses several tasks, including morphological segmentation, morphological analysis, and disambiguation. It serves as a foundational step in many NLP applications, including information retrieval, word sense disambiguation (Jarrar et al., 2023), machine translation (McGill et al., 2023), and named-entity recognition (Darwish et al., 2014).

Arabic morphological tagging is challenging due to the language's rich inflectional and derivational morphology, where prefixes and suffixes encode features such as gender, number, tense, and aspect (Ryding, 2014; Pasha et al., 2014). The task becomes even more complex in Arabic dialects because of dialect-specific affixes, POS tags and lemmas (Pasha et al., 2014). Although substantial progress has been achieved for MSA through large-scale resources and pre-trained language models (Inoue et al., 2021; Darwish et al., 2020), dialectal Arabic morphology remains largely under-resourced, and constructing multidialectal morphologically annotated corpora is still costly and labor-intensive.

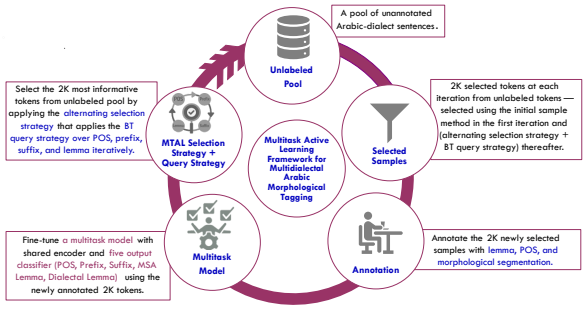

Figure 1: Multitask Active learning framework for multidialectal Arabic Morphological tagging

Active learning reduces annotation costs by prioritizing the most informative samples and has shown effectiveness in single-task settings such as POS tagging (Akra et al., 2025). However, its role in multitask morphological tagging remains unclear, particularly when jointly learning tasks such as POS tagging, lemmatization, and segmentation, where identifying informative tokens across multiple tasks is challenging. Another challenge in active learning is the need to repeatedly fine-tune the language model at each iteration. As the labeled dataset grows over successive iterations, the computational cost of retraining increases substantially. This motivates the use of continual learning, which enables incremental model updates while preserving previously acquired knowledge (Das

et al., 2023).

Our contributions are summarized as follows:

- Evaluation of multitask active learning selection strategies (seven experiments), which show that the *alternating selection* strategy outperforms the others, converging at an average of $16,000$ tokens across tasks.
- Evaluation of continual active learning methods (30 experiments), proving that the *Uncertainty-Common* method successfully addresses catastrophic forgetting. It outperforms Experience Replay baseline by $\sim$16% in POS tagging F_1-score while saving $\sim$2.5 minutes per iteration relative to the full re-finetuning.
- Multitask active learning framework for multidialectal Arabic morphological tagging (Figure 1), which we evaluated on the combined Nabra (Syrian) and Baladi (Lebanese), converging on average at $16,000$ tokens and achieving 91.15%, 95.71%, and 90.45% for POS, prefix, and suffix tagging, respectively.

The paper is structured as: Section 2 reviews related work; Section 3 describe the corpora, Sections 4, 5 describe the continual multitask active learning methodology; Sections 6 and 7 present results and findings; and we conclude in Section 8.

2 Related Work

A range of methods has been employed for Arabic and dialectal morphological tagging. Some approaches focus on a single morphology task (Darwish et al., 2017, 2018), while others handle multiple morphology tasks jointly, training the tasks together (Pasha et al., 2014; Kondratyuk, 2019; Inoue et al., 2022). Additional approaches address multiple morphology tasks without joint learning, treating each task independently (Boudchiche et al., 2017; Buckwalter, 2004; Maamouri et al., 2010; Jarrar et al., 2024).

All of the aforementioned machine learning and deep learning approaches rely on passive learning, which requires large annotated datasets for training. Most existing methods are designed primarily for MSA, with only a few tailored to dialects—mainly Egyptian and Levantine. Consequently, developing a multidialectal morphology tagger demands a large morphologically annotated dataset.

Active learning offers a promising direction for building a multidialectal POS tagger using only a minimal set of highly informative annotated tokens (Akra et al., 2025). However, none of the previously mentioned approaches that employ multitask learning for morphological tagging have explored active learning in a multitask (joint) setting.

In the context of active learning, (Rotman and Reichart, 2022) investigated multitask (joint) active learning (MTAL) for several types of task relations: complementary tasks such as dependency parsing and named entity recognition, hierarchical tasks such as relation extraction and named entity recognition, and even two unrelated tasks such as slot filling and intent detection. They introduced several MTAL selection strategies, including: (1) One-sided selection, where new examples are chosen based on a single task; (2) Alternating selection, where selection alternates between tasks across iterations; (3) Aggregation (joint) selection, where examples are selected based on a combined scoring function across all tasks.

To the best of our knowledge, multitask active learning has not yet been explored in the context of morphological tagging.

Another important aspect of active learning is the re-finetuning time required at each iteration, which increases substantially as the annotated dataset grows. This escalating cost can become prohibitive, as the model is typically re-finetuned from scratch after each active learning iteration. In this context, continual learning, where the model is incrementally fine-tuned on newly selected samples at each iteration, presents a promising solution.

Although several continual learning approaches have been proposed in the literature, only a limited number have been explored in conjunction with active learning. In other words, few studies have investigated the use of continual learning methods to mitigate the increasing re-finetuning time associated with active learning. Such study is (Das et al., 2023), which evaluated several continual learning techniques within an active learning framework, including experience replay, dark experience replay, maximally interfered retrieval, scaled distillation, and scaled distillation with submodular sampling.

To the best of our knowledge, continual active learning has not yet been explored in the context of morphological tagging. Consequently, this work is the first to introduce an approach that combines active learning, multitask learning, and continual learning, resulting in a continual multitask active learning framework for morphological tagging.

3 Datasets Preparation and Composition

This paper investigates three Arabic dialects—Palestinian, Syrian, and Lebanese, using three datasets that fully annotated with morphological features (POS, MSA Lemma, Dialect Lemma, Prefix, Stem, Suffix, and others): (1) Curras (Palestinian) (Haff et al., 2022), (2) Nabra (Syrian) (Nayouf et al., 2023), (3) Baladi (Lebanese) (Haff et al., 2022) [1]. Table 1 summarizes these datasets.

Dataset	Tokens	Unique				
		Tokens	Lemma	POS	Prefix	Suffix
Curras	56,700	16,573	6,148	43	178	504
Nabra+Baladi	69,582	24,664	8,715	46	248	473

Table 1: Datasets Statistics

The datasets were split following the methodology in (van der Goot, 2021): TRAIN for training, TUNE (5% of TRAIN) for model selection, DEV for initial evaluation, and TEST for final evaluation. Table 2 shows the token counts for TRAIN, DEV, and TEST. Some tokens were excluded due to missing annotations in the original datasets.

Dataset	TRAIN	DEV	TEST
Curras	44,643	5,708	5,816
Nabra + Baladi	56,174	6,460	6,948

Table 2: Splits statistics for each dataset

4 Multi-Task Active Learning

In this section we present a multitask active learning framework for morphological tagging tasks. We first introduce a multitask learning architecture for morphological tagging (Section 4.1). We then evaluate different multitask active learning strategies to identify the most effective approach across all morphological tagging tasks (Section 4.2).

In our methodology, the active learning process continues until all tokens in each corpus's training set are annotated. This allows us to track the impact of each newly selected batch of tokens at every iteration and for each task, and to evaluate how increasing annotation coverage affects performance in each morphological tagging task.

4.1 Multi-Task Active Learning Architecture

In multi-tasking we will learn multiple tasks simultaneously so that each task benefits from shared information. Motivated by this, we developed a multitask learning architecture for morphological tagging, covering POS tagging, morphological segmentation (prefix and suffix tagging), and lemmatization (MSA and dialectal).

All layers except the output layers are shared across tasks. Our architecture includes five classifiers: POS, prefix, suffix, MSA lemma, and dialectal lemma. Each classifier has its own loss function, and during training, the losses from all five outputs are aggregated to optimize the model.

We use this architecture within an active learning setting, introducing a multitask active learning framework. In our framework, we adopt *Breaking Ties* as the query strategy, which is a prediction-based method that selects tokens with the smallest margin between the top two predicted labels for annotation. For the initial sample selection, we use the *Most Dissimilar Sentences* method, which identifies sentences from the unlabeled pool that are maximally dissimilar, up to a limit of 2,000 tokens in the first iteration. The learning model is *AraBERTV2*, a pretrained language model trained on both MSA and dialectal Arabic, following (Akra et al., 2025). (Akra et al., 2025) showed that Breaking Ties converges faster than other strategies for POS tagging and the Most Dissimilar Sentences yields superior performance at the initial iteration, and AraBERTV2 consistently surpasses other Arabic pretrained models.

Although our model is trained in a multitask active learning setting, a key question concerns which task the breaking ties strategy should be applied to. This issue is addressed through different multitask active learning strategies, which are presented in Section 4.2 and described in detail therein.

4.2 Multi-Task Active learning Strategies

In multitask (joint) active learning, we start with a corpus that contains initial tokens that are morphologically annotated for various morphology tasks, which include lemmatization, POS tagging, and segmentation into prefixes and suffixes. The process is iterative: at the k_{th} iteration, the goal is to select L_{new} additional unlabeled samples for annotation across all morphological tagging tasks, aiming to maximize the overall model improvement in these tasks. As stated earlier, Breaking Ties is used as the query strategy to select new samples. The question is which task or combination of tasks the query strategy is applied to; this defines the multitask active learning strategies (Rotman and Reichart, 2022), such as:

[1] Curras, Baladi, Nabra are CC-BY-4.0

- **Single task selection**: In this strategy, all tasks are engaged in the training process; however, only one task is utilized to select new samples during each iteration. Specifically, a multitask model is trained on all morphological tagging tasks and generates predictions for all tasks over the unlabeled pool; however, sample selection is driven by the predictions of a single task. For example, tokens are selected based on the minimal difference between the top two POS predictions. In our methodology, we perform single-task selection using POS tagging and evaluate performance across all tasks at each iteration. The same procedure is applied separately using lemmatization, prefix tagging, and suffix tagging as the selection task.

- **Alternating selection**: In this strategy, the Breaking Ties query strategy selects new tokens for annotation based on the predictions of a single task per iteration, while performance is evaluated across all tasks at each iteration. Unlike the previous strategy, the selection task is not fixed but alternates between iterations. In our methodology, selection is based on POS tagging in the first iteration, lemmatization in the second, prefix tagging in the third, and suffix tagging in the fourth, repeating this cycle until the final iteration.

- **Aggregation(Joint) selection**: This strategy differs from the previous two in that Breaking Ties selects new tokens based on predictions from multiple tasks rather than a single one. Breaking Ties scores are computed for each task $i \in T$ (e.g., T = [POS tagging, Lemmatization, Prefix Tagging, Suffix Tagging]) for each token x in the unlabeled pool U, then aggregated across tasks using an aggregation function (e.g., averaging), and the tokens with the minimal aggregated Breaking Ties scores will be selected for annotation, formally a token x is selected from unlabeled pool U with:

$$\arg\min_{x \in U} \frac{1}{|T|} \sum_{i \in T} \left(P(y^i_{l1}|x) - P(y^i_{l2}|x) \right)$$

  where T represents the set of morphological tasks, $|T|$ denotes the total number of tasks (the magnitude of the set), and (y^i_{l1}, y^i_{l2}) correspond to the first- and second-most probable labels assigned by the model to token x specifically for task i.

5 Continual Multi-Task Active Learning

This section describes the methodology used to train the proposed multitask active learning model in Section 4 in a continual manner. Unlike traditional active learning, where the model is retrained from scratch at each iteration—leading to rapidly increasing training time, especially in multitask settings—we adopt a **continual learning** approach that incrementally updates the model with newly selected data. However, continual learning introduces the risk of **catastrophic forgetting**, where previously learned knowledge is lost when training on new data. In Section 5.1 we describe the methodology of training the multitask active learning model in a continual manner, while in Section 5.2 we introduce the different continual learning methods that we evaluated.

5.1 Continual Learning Methodology

In our methodology, we integrate continual learning into the multitask active learning framework by updating the model at each iteration with newly selected tokens and samples from previous iterations. The methodology is governed by a Training Budget (TB), which defines the maximum number of tokens permitted for training at each iteration.

The methodology follows these phases: We begin at **Iteration** $k = 0$ with an initial set of morphologically annotated tokens. We fine-tune the multitask model on this initial set, resulting in a model with parameters θ_0. At **Iteration** k, we select new tokens from the unlabeled pool U using the best-performing multitask active learning selection method (Alternating Selection; see Section 6.2). After annotation, the multitask model with parameters θ_{k-1} update follows one of two paths based on the total count of annotated tokens (L): For **Full Re-finetuning**, if the total number of cumulative annotated tokens (L) is less than the training budget (TB), the model is re-initialized and fine-tuned on all annotated tokens up to that iteration. For the **Continual Update**, if the number of annotated tokens (L) exceeds the training budget (TB), the model (θ_{k-1}) is updated continually on the newly selected batch while preserving knowledge from previous iterations. This process results in a trained model with parameters θ_k.

5.2 Continual Learning Methods

To preserve knowledge efficiently, we maintain an **Uncertainty-Cached History Buffer** (H). Unlike

standard history buffers that store only raw data, we cache the Breaking Ties scores generated during the active learning iterations. This allows our continual learning methods to prioritize samples for B_{replay} without requiring expensive forward passes over the entire history.

In addition to a baseline **Experience Replay** approach, which selects B_{replay} via random sampling from H, we evaluate several continual learning methods. These methods are defined by a general priority score $S(w)$ for each word $w \in H$:

$$S(w) = \alpha \cdot \phi(w) + \beta \cdot \psi(w) \quad (1)$$

Here, $\phi(w)$ represents the **Confidence Factor** (derived from cached breaking ties (BT) scores) and $\psi(w)$ represents the **Representativeness Factor** (derived from frequency analysis). The α and β are used for weighting these factors.

To prevent redundancy in B_{replay}, we employ **Submodular Selection**. We greedily maximize an objective function $G(\mathcal{S})$ that balances the word-level priority $S(w)$ with sentence-level coverage:

$$G(\mathcal{S}) = \sum_{w \in \mathcal{S}} S(w) + \sum_{sid \in \text{Sent}(\mathcal{S})} \mathbb{I}(sid \notin C) \quad (2)$$

where C denotes the set of sentences already included in the buffer. Table 3 summarizes these methods.

Method	Confidence $\phi(w)$	Frequency $\psi(w)$
Heuristic Selection (Top-N Sorting)		
Common Freq.	–	$\ln(1 + \text{freq}_{\text{norm}})$
Rare Freq.	–	$-\ln(1 + \text{freq}_{\text{norm}})$
Max BT (Certainty)	$BT(w)$	–
Min BT (Uncertainty)	$1 - BT(w)$	–
Certainty-Common	$BT(w)$	$\ln(1 + \text{freq}_{\text{norm}})$
Certainty-Rare	$BT(w)$	$-\ln(1 + \text{freq}_{\text{norm}})$
Uncertainty-Common	$1 - BT(w)$	$\ln(1 + \text{freq}_{\text{norm}})$
Uncertainty-Rare	$1 - BT(w)$	$-\ln(1 + \text{freq}_{\text{norm}})$
Submodular Selection (Diversity-Aware)		
Certainty (Sub)	$BT(w)$	–
Uncertainty (Sub)	$1 - BT(w)$	–
Certainty-Common (Sub)	$BT(w)$	$\ln(1 + \text{freq}_{\text{norm}})$
Certainty-Rare (Sub)	$BT(w)$	$-\ln(1 + \text{freq}_{\text{norm}})$
Uncertainty-Common (Sub)	$1 - BT(w)$	$\ln(1 + \text{freq}_{\text{norm}})$
Uncertainty-Rare (Sub)	$1 - BT(w)$	$-\ln(1 + \text{freq}_{\text{norm}})$

Table 3: Continual Learning Methods. Note: $BT(w)$ refers to cached Breaking Ties scores; $\text{freq}_{\text{norm}}$ is the normalized word frequency in history H.

As a summary, Algorithm 1 describes the complete workflow of the proposed framework, integrating multitask, active and continual learning.

6 Evaluation Results

This section presents the evaluation results based on the methodology outlined in Sections 4, and 5, the selection of the best multitask active learning strategy (6.2), the selection of best continual active learning method (6.3).

Algorithm 1 Continual Multitask Active Learning

Input: Unlabeled Pool U, Task List $T = [POS, Lemma, Prefix, Suffix]$, Training Budget TB, Hyper-parameters α, β
Output: Fine-tuned Multitask Model θ_K
$H \leftarrow \emptyset$ {Initialize Uncertainty-Cached History Buffer}
$L \leftarrow \emptyset$ {Initialize Labeled Pool}
for iteration $k = 0$ to K **do**
  {**Phase 1: Active learning Selection Strategy**}
  if $k = 0$ **then**
    $L_{new} \leftarrow$ most_dissimilar_sentences(U)
    $B_{replay} \leftarrow \emptyset$ {No history exists at $k = 0$}
  else
    {**Phase 1.1: Multitask Active learning Selection Strategy**}
    $t \leftarrow$ Select task from T (Alternating Strategy)
    $S \leftarrow \{x \in U \mid$ minimal Breaking Ties scores for task $t\}$
    $L_{new} \leftarrow$ Annotate(S)
    if $|L| > TB$ **then**
      {**Phase 1.2: Selection from History**}
      Compute $S(w) = \alpha\phi(w) + \beta\psi(w)$ for all $w \in H$
      $B_{replay} \leftarrow$ continual_learning_method($H, S(w)$, size $= TB - |L_{new}|$)
    end if
  end if
  {**Phase 3: Continual Model Update**}
  if $|L| < TB$ **then**
    $\theta_k \leftarrow$ Fine-tune(θ_{base}, L) {Full Re-finetuning}
  else
    $\theta_k \leftarrow$ Fine-tune($\theta_{k-1}, L_{new} \cup B_{replay}$) {Continual Update}
  end if
  {**Phase 4: Pools (labeled, Unlabeled) & History Buffer Update**}
  $L \leftarrow L \cup L_{new}$
  $U \leftarrow U \setminus L_{new}$
  $H \leftarrow H \cup \{(L_{new}, BT(L_{new}))\}$ {Cache BT scores with tokens}
end for
return θ_K

6.1 Experimental Setup

The implementation of all models utilized the Hugging Face Transformers library (Wolf et al., 2020) and trained in PyTorch (Paszke et al., 2019). Consistent experimental conditions were maintained across all experiments unless otherwise specified; each active learning iteration comprised 10 epochs with a fixed learning rate of 5e−5, a batch size of 6, and a random seed of 12345. The total number of active learning iterations was scaled proportionally to the size of the training set in each corpus.

For the proposed continual active learning methods, we define specific hyper-parameters to balance confidence and representativeness. Specifically, we set the confidence weight $\alpha = 0.7$ and the representativeness weight $\beta = 0.3$. This configuration prioritizes model uncertainty while maintaining a significant signal for word frequencies in the replay buffer.

All Experiments were run on a server with 62GiB RAM, 1.2TB disk storage, and a two NVIDIA T4 GPUs. All experiments are conducted on the Curras (Palestinian) dataset, unless stated otherwise.

6.2 Multi-Task Active Learning Results

Table 4 summarizes the F_1-scores of each multitask active learning selection strategy across all tasks at

key milestones: the second iteration ($4K$ tokens), the stabilization point ($16K$ tokens), and the final iteration ($42K$ tokens). Figure 2 presents a detailed iteration-by-iteration performance for POS tagging, while the corresponding figures for the remaining tasks are provided in Appendix A.

Task	MTAL Selection	$4K$	$16K$	$42K$
POS Tagging	Single task (baseline)	85.22	93.50	95.09
	Alternating	**88.86**	94.65	**95.18**
	Aggregation	83.94	**94.66**	95.14
Lemmatization	Single task (baseline)	32.73	59.94	75.22
	Alternating	**38.13**	**63.53**	75.50
	Aggregation	27.94	58.84	**75.54**
Prefix Tagging	Single task (baseline)	92.30	96.95	97.29
	Alternating	92.16	**97.61**	**97.28**
	Aggregation	**93.94**	97.32	**97.28**
Suffix Tagging	Single task (baseline)	85.79	92.57	94.21
	Alternating	85.71	**93.33**	**94.13**
	Aggregation	**86.67**	93.18	94.09

Table 4: MTAL strategies results.

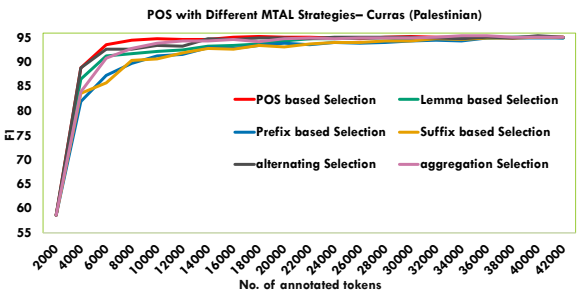

Figure 2: POS Performance on MTAL strategies

6.3 Continual Multi-Task Active Learning Results

Table 5 reports the performance of the best-performing heuristic and submodular continual methods compared with both baselines at the final active learning iteration (21) across all tasks.

To track performance across active learning iterations, Table 6 presents POS tagging results at key active learning iterations under a fixed training budget ($TB = 10{,}000$ tokens). Iteration 5 serves as the anchor point, where the model reaches the maximum number of tokens defined by the TB. From Iteration 6 onward, continual learning mechanisms are activated to manage history preservation while active learning continues to query new data sequentially. Performance across active learning iterations for lemmatization, prefix, and suffix tagging are provided in Appendix C.

Complementing the performance matrix,Table 7 presents the average iteration time incurred by each method,which is the average replay selection time and the average training time.

Appendix D presents the evaluation results of the continual learning methods under another training budget setting ($TB = 20{,}000$).

7 Discussion

7.1 Best Multitask Active Learning Selection Strategy

Based on Figure 2, it is notable that POS-based selection achieves the best performance for the POS task, and the same trend is observed for the remaining tasks (see Appendix A). In other words, single-task selection performs best only on its corresponding task. However, this strategy does not maintain strong performance across all tasks. As shown in Table 4, the single-task setting—represented by the average results of POS-based, lemma-based, prefix-based, and suffix-based selection— is such that the overall average remains lower because its high performance is limited strictly to its corresponding task. For example, the accuracy of Suffix tagging at the final iteration using Suffix-based selection reaches 94.33%, while its performance on other tasks is the lowest, leading to the lowest overall average.

Since multitask active learning aims to achieve balanced performance across tasks, we consider single-task selection as a baseline and focus on comparing alternating and aggregation strategies, which show only minor performance differences. Specifically, as shown in Table 4, alternating slightly outperforms aggregation for all tasks. Here, "outperforming" refers to faster convergence to higher performance, meaning that a strategy reaches better accuracy at earlier iterations.

As a result, the proposed multitask active learning framework, using the best-performing selection strategy (alternating selection), was evaluated on the combined Nabra (Syrian) and Baladi (Lebanese) corpora. The two corpora were merged to construct a larger multidialectal corpus.

Figure 3 shows that POS, prefix, and suffix tagging begin to stabilize at approximately 10K tokens, whereas lemmatization stabilizes after a larger number of tokens (around 22K). This is due to the higher complexity of lemmatization, which is treated as a classification task with a large label space (approximately 6K classes), making learning more difficult and requiring more tokens to capture class-specific patterns. Therefore, the overall stabilization point is set to approximately 16K tokens on average. At this point, the framework achieves

Task	Baselines		Top Heuristic Selection		Top Submodular Selection	
	Without Continual	Experience Replay	Strategy	F_1	Strategy	F_1
POS tagging	95.18	73.83	Uncertainty-Common	89.39	Uncertainty-Common (Sub)	**89.85**
Lemmatization	75.50	56.48	Certainty-Rare	47.30	Certainty-Rare (Sub)	**50.97**
Prefix Tagging	97.28	77.85	Min BT (Uncertainty)	95.54	Uncertainty (Sub)	**96.38**
Suffix Tagging	94.13	77.35	Rare Freq.	**88.88**	Certainty-Rare (Sub)	88.12

Table 5: Final-iteration performance of top heuristic and submodular methods ($TB = 10,000$).

Method	It.5	It.7	It.14	It.21
Baselines				
without Continual Learning	93.43	94.85	95.07	95.18
Experience Replay	93.43	75.98	77.37	73.83
Heuristic Selection (Top-N Sorting)				
Common Freq.	93.43	**94.77**	90.18	29.59
Rare Freq.	93.43	87.21	78.58	77.86
Max BT (Certainty)	93.43	91.54	91.48	64.05
Min BT (Uncertainty)	93.43	94.21	76.30	87.55
Certainty-Common	93.43	91.30	**91.63**	33.95
Certainty-Rare	93.43	89.38	88.79	85.67
Uncertainty-Common	93.43	94.35	89.10	**89.39**
Uncertainty-Rare	93.43	94.68	77.40	77.75
Submodular Selection (Diversity-Aware)				
Certainty (Sub)	93.43	92.48	**92.04**	86.16
Uncertainty (Sub)	93.43	94.59	91.83	89.60
Certainty-Common (Sub)	93.43	92.42	91.46	84.86
Certainty-Rare (Sub)	93.43	92.30	91.83	87.28
Uncertainty-Common (Sub)	93.43	**94.96**	90.92	**89.85**
Uncertainty-Rare (Sub)	93.43	94.38	91.16	85.89

Table 6: POS Performance on Different Continual Learning Methods at TB $(10,000)$ across Iterations.

Method	Selection Time	Training Time	Total Time
Baselines			
without Continual Learning	00.00	13.79	13.79
Experience Replay	0.001	11.70	11.70
Heuristic Selection (Top-N Sorting)			
Common Freq.	00.14	11.76	11.90
Rare Freq.	00.12	11.71	11.83
Max BT (Certainty)	00.09	11.30	11.39
Min BT (Uncertainty)	00.09	11.60	11.69
Certainty-Common	00.12	10.82	10.94
Certainty-Rare	00.12	13.34	13.46
Uncertainty-Common	00.12	11.26	11.38
Uncertainty-Rare	00.12	12.06	12.18
Submodular Selection (Diversity-Aware)			
Certainty (Sub)	00.76	13.64	14.40
Uncertainty (Sub)	00.77	13.58	14.35
Certainty-Common (Sub)	00.82	13.68	14.50
Certainty-Rare (Sub)	00.82	13.57	14.39
Uncertainty-Common (Sub)	00.82	13.66	14.48
Uncertainty-Rare (Sub)	00.81	13.63	14.44

Table 7: Average Time Performance (Minutes) on Different Continual Learning Methods at TB $(10,000)$.

91.15%, 95.71%, and 90.45% for POS, prefix, and suffix tagging, respectively, as shown in Figure 3 (for further details, see Table 8 in Appendix B).

Notably, the POS tagging F1 score at convergence (16,000 tokens), namely 91.5%, slightly under performs the 92.9% reported by (Akra et al., 2025). This difference arises because our experiments were conducted in a multitask active learning setting, where the model was trained jointly on multiple tasks, whereas (Akra et al., 2025) evaluated active learning on a single task only, namely POS tagging. Therefore, the reported scores are not directly comparable.

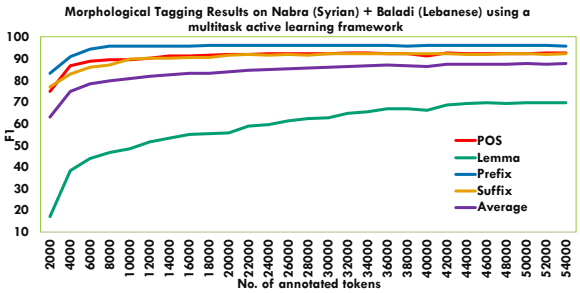

Figure 3: Morphological Tagging Results on Nabra (Syrian) + Baladi (Lebanese).

7.2 Best Continual Active Learning Method

Table 5 indicates that continual learning methods generally reduce catastrophic forgetting compared to the random baseline (Experience Replay), although effectiveness varies across tasks. The most severe degradation is observed in lexical tasks such as lemmatization (56.48%) and morphologically dense tasks such as suffix tagging (77.35%), indicating that random replay is insufficient for preserving rare or fine-grained morphological inflectional patterns.

While continual learning methods generally mitigate catastrophic forgetting, the optimal method is task dependent. For POS and prefix tagging, uncertainty-based methods perform best, with *Uncertainty-Common (Sub)* and *Uncertainty (Sub)* achieving 89.85% and 96.38%, respectively. This is because the main challenge in POS and prefix tagging is ambiguity, so prioritizing ambiguous tokens leads to better performance. For lemmatization, a sparse lexical task, *Certainty-Rare (Sub)* achieves the best result (50.97%), indicating that focusing on confident but infrequent tokens better supports lexical stability than uncertainty-based sampling. In contrast, suffix tagging benefits most from *Rare Freq.* (88.88%), suggesting that emphasizing rare morphological inflectional patterns is more effective than applying diversity in highly inflected settings. Overall, submodular diversity methods provide consistent improvements across most tasks, with an average gain of approximately

1.5% over heuristic methods.

Table 6 shows that traditional replay baselines suffer severe degradation under multitask continual learning. Standard *Experience Replay* exhibits catastrophic forgetting once continual learning begins at Iteration 6, with performance dropping from 93.43% to 75.98% at Iteration 7 and further declining to 73.83% by Iteration 21. This indicates that random replay sampling is insufficient to preserve prior knowledge across successive active learning iterations.

A similar pattern is observed in top-N heuristic methods that lack diversity constraints. For example, the *Common Freq.* method shows strong short-term stability during early iterations, reaching 94.77% at Iteration 7. This is due to the active learning query strategy initially selecting highly informative dialectal words (e.g., ولاد / *wlād* at Iteration 1 and شو / *šw* at Iteration 5), which dominate the early historical buffer when continual learning begins at Iteration 6. However, as the history pool expands, frequent tokens become increasingly dominated by common functional words such as من / *mn* and على / *ɬā* . Consequently, previously learned dialectal features are gradually underrepresented, leading to catastrophic forgetting and a sharp decline to 29.59% by Iteration 21. In contrast, the *Uncertainty-Common* strategy maintains performance by prioritizing frequent historical tokens for which the model remains uncertain, thereby preserving critical dialectal knowledge and achieving a final performance of 89.39%.

The proposed diversity-aware submodular methods provide stronger resistance to forgetting. The best-performing *Uncertainty-Common (Sub)* achieves 94.96% at Iteration 7, slightly outperforming the *without continual learning* baseline (94.85%), suggesting that diversity-maximizing selection yields more informative replay samples than standard heuristics. By Iteration 21, all submodular methods maintain stable performance, with *Uncertainty-Common (Sub)* preserving 89.85%.

Table 7 highlights the computational trade-off between heuristic and submodular methods. Submodular selection introduces higher overhead due to greedy subset optimization over an expanding history pool, requiring approximately 0.82 minutes compared to 0.12 minutes for heuristic sorting. Training time also increases (about 13.6 minutes versus 11.7 minutes for standard replay) because diversity-aware selection distributes the fixed token budget across more unique sentences, increasing mini-batch iterations and GPU padding overhead.

Overall, although submodular methods achieve the highest knowledge preservation for three out of four tasks, their total iteration time (14.48 minutes) exceeds that of the *without continual learning* re-finetuning baseline (13.79 minutes), which conflicts with the objective of reducing re-finetuning time. Therefore, the best-performing heuristic method for each task is selected as the preferred continual learning strategy, as it offers a better balance between performance and efficiency (e.g., for POS tagging: 89.39% and 11.38 minutes).

Finally, although a performance degradation is observed across all tasks (ranging from 1 point in prefix tagging to 25 points in lemmatization compared to the without continual learning), the ability to save approximately 2.5 minutes per iteration relative to the non-continual learning setup may be significant in certain scenarios. In other words, in time-critical settings, reducing training time can be more important than marginal performance losses.

Generalized Methodological Insights: An overall finding of our framework is that optimal data selection criteria are task-dependent. Tasks that benefit from ambiguity, such as POS and prefix tagging, consistently perform better with uncertainty-based selection (min_BT). Conversely, highly inflected and sparse lexical tasks (e.g., lemmatization and suffix tagging) benefit more from certainty-based selection (max_BT). This task dependency indicates that multidialectal active learning frameworks cannot rely on a single selection objective; instead, multi-task setups must strategically alternate or balance uncertainty and certainty to align with the requirements of each morphological tagging task.

8 Conclusion

We proposed a continual multitask active learning framework for multidialectal Arabic morphological tagging. Our evaluation shows that an alternating selection strategy querying approximately $16{,}000$ informative tokens achieves high performance across tasks. Furthermore, integrating continual learning via the *Uncertainty-Common* method reduces iteration training time by roughly 2.5 minutes while effectively minimizing catastrophic forgetting. These findings demonstrate that our framework significantly lowers annotation costs for dialectal Arabic morphological tagging.

9 Limitations

Despite its effectiveness, the proposed continual multitask active learning framework has several limitations. First, the evaluation is restricted to specific tasks and datasets, which may limit the generalizability of the findings. Second, although the framework is well suited for standard active learning cycles, its long-term scalability is constrained by the fixed parameter capacity of the underlying pre-trained language model (AraBERTV2). While AraBERTV2 provides strong contextual representations for multidialectal Arabic, compact models inherently have limited capacity for storing new structural patterns. Consequently, under extremely long or unconstrained continuous annotation settings, the model may eventually encounter capacity saturation.

References

Diyam Akra, Mohammed Khalilia, and Mustafa Jarrar. 2025. Active Learning for Multidialectal Arabic POS Tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24960–24973, Suzhou, China. Association for Computational Linguistics.

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*, 29:141–146.

Tim Buckwalter. 2004. Buckwalter arabic morphological analyzer version 2.0. Web download. LDC Catalog No. LDC2004L02.

Kareem Darwish, Ahmed Abdelali, and Hamdy Mubarak. 2014. Using stem-templates to improve Arabic POS and gender/number tagging. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2926–2931, Reykjavik, Iceland. European Language Resources Association (ELRA).

Kareem Darwish, Mohammed Attia, Hamdy Mubarak, Younes Samih, Ahmed Abdelali, Lluís Màrquez, Mohamed Eldesouki, and Laura Kallmeyer. 2020. Effective multi-dialectal arabic pos tagging. *Natural Language Engineering*, 26:677–690.

Kareem Darwish, Hamdy Mubarak, Ahmed Abdelali, and Mohamed Eldesouki. 2017. Arabic pos tagging: Don't abandon feature engineering just yet. *Proceedings of The Third Arabic Natural Language Processing Workshop (WANLP)*, pages 130–137.

Kareem Darwish, Hamdy Mubarak, Ahmed Abdelali, Mohamed Eldesouki, Younes Samih, Randah Alharbi, Mohammed Attia, Walid Magdy, and Laura Kallmeyer. 2018. Multi-dialect Arabic POS tagging: A CRF approach. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Arnav Das, Gantavya Bhatt, Megh Bhalerao, Vianne Gao, Rui Yang, and Jeff Bilmes. 2023. Accelerating batch active learning using continual learning techniques. *Preprint*, arXiv:2305.06408.

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + Baladi: Towards a Levantine Corpus. In *Proceedings of the International Conference on Language Resources and Evaluation(LREC 2022)*, Marseille, France.

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.

Mustafa Jarrar, Diyam Akra, and Tymaa Hammouda. 2024. ALMA: Fast Lemmatizer and POS Tagger for Arabic. In *Proceedings of the 2024 AI in Computational Linguistics (ACLING 2024)*, Procedia Computer Science, Dubai. ELSEVIER.

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammed Khalilia. 2023. SALMA: Arabic Sense-annotated Corpus and WSD Benchmarks. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*, pages 359–369. ACL.

Dan Kondratyuk. 2019. Cross-lingual lemmatization and morphology tagging with two-stage multilingual BERT fine-tuning. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 12–18, Florence, Italy. Association for Computational Linguistics.

Mohamed Maamouri, David Graff, Basma Bouziri, Sondos Krouna, Ann Bies, and Seth Kulick. 2010. Standard arabic morpholgical analyzer(sama) version 3.1. Web download. LDC Catalog No. LDC2010L01.

Euan McGill, Luis Chiruzzo, Santiago Egea Gómez, and Horacio Saggion. 2023. Part-of-speech tagging {S}panish {S}ign {L}anguage data and its applications in sign language machine translation. *Proceedings of the Second Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL-2023)*, pages 70–76.

Amal Nayouf, Tymaa Hammouda, Mustafa Jarrar, Fadi Zaraket, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian Arabic Dialects with Morphological Annotations. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*, pages 12–23. ACL.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 1094–1101. European Language Resources Association (ELRA).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA.

Guy Rotman and Roi Reichart. 2022. Multi-task active learning for pre-trained transformer-based models. *Transactions of the Association for Computational Linguistics*, 10:1209–1228.

Karin C. Ryding. 2014. *Arabic: A Linguistic Introduction*. Cambridge University Press.

Rob van der Goot. 2021. We need to talk about train-dev-test splits. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4485–4494, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Multi-Task Active learning Detailed Results

Figures 2, 4, 5, 6 show performance on other morphological tagging tasks (lemmatization, prefix tagging, and suffix tagging) on the Curras (Palestinian) corpus, respectively, using different multitask active learning strategies. The strategies (POS based, Lemma based, Prefix based, and Suffix based) are all categorized under single task selection, so in those experiments, the breaking ties query strategy is applied over POS, Lemma, Prefix, and Suffix probabilities, respectively, as we described previously in methodology 4.

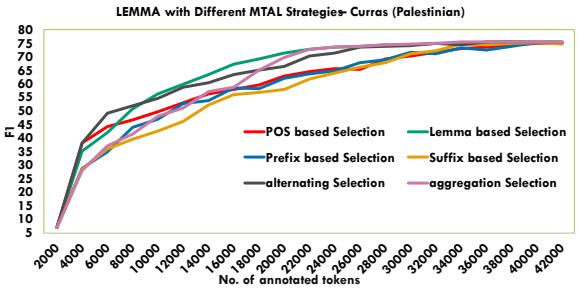

Figure 4: Lemma Performance on Different Multitask Active learning strategies

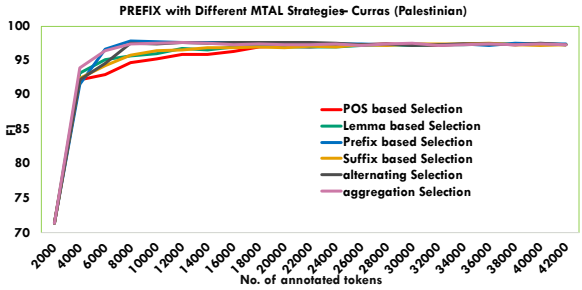

Figure 5: Prefix Performance on Different Multitask Active learning strategies

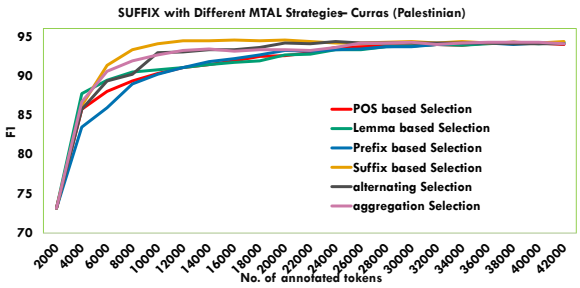

Figure 6: Suffix Performance on Different Multitask Active learning strategies

B Evaluation of Best Multi-task active learning on Nabra+Baladi

Table 8 presents the performance of the proposed multitask active learning framework using the alternating selection strategy on the combined Nabra+Baladi corpus.

Tokens	POS	Lemma	Prefix	Suffix	Average
$2K$	74.93	17.19	83.28	76.82	63.06
$4K$	86.75	38.27	90.90	82.88	74.70
$6K$	88.71	44.03	94.16	86.13	78.26
$8K$	89.45	46.66	95.58	86.87	79.64
$10K$	89.31	48.55	95.55	89.88	80.82
$12K$	90.05	51.40	95.62	90.27	81.84
$14K$	91.24	53.17	95.77	90.16	82.59
$16K$	91.15	55.05	95.71	90.45	83.09
$18K$	91.63	55.33	95.98	90.44	83.35
$20K$	91.88	55.83	96.00	91.58	83.82
$22K$	91.92	58.70	95.92	91.75	84.57
$24K$	92.12	59.68	96.05	91.61	84.87
$26K$	92.32	61.24	96.08	91.72	85.34
$28K$	92.27	62.32	96.18	91.59	85.59
$30K$	92.41	62.77	96.21	92.28	85.92
$32K$	92.53	64.73	96.09	92.18	86.38
$34K$	92.48	65.34	96.10	92.16	86.52
$36K$	92.38	66.93	96.06	92.09	86.87
$38K$	92.30	66.97	95.82	92.09	86.80
$40K$	91.29	66.15	96.00	92.07	86.38
$42K$	92.42	68.68	96.03	92.13	87.32
$44K$	92.33	69.23	95.96	92.02	87.39
$46K$	92.41	69.57	95.92	91.94	87.46
$48K$	92.40	69.46	95.96	92.28	87.53
$50K$	92.31	69.73	96.05	92.12	87.55
$52K$	92.49	69.65	95.91	92.02	87.52
$54K$	92.51	69.76	95.84	92.10	87.55

Table 8: Morphological Tagging Results on Nabra (Syrian) + Baladi (Lebanese) using a multitask active learning framework. Tokens here means No. of Annotated Tokens

C Continual Learning Results for other tasks at budget 10,000

Tables 9, 10, and 11 report the performance of prefix tagging, suffix tagging, and lemmatization, respectively, across key active learning iteration milestones (5, 7, 14, and 21) under a fixed training budget (TB) of 10,000 tokens for each continual learning method.

Method	It.5	It.7	It.14	It.21
Baselines				
without Continual Learning	54.73	60.56	74	75.50
Experience Replay	54.73	53.82	58.42	56.48
Heuristic Selection (Top-N Sorting)				
Common Freq.	54.73	**60.45**	58.30	24.63
Rare Freq.	54.73	39.90	31.93	27.33
Max BT (Certainty)	54.73	52.15	58.09	28.85
Min BT (Uncertainty)	54.73	48.48	38.28	31.07
Certainty-Common	54.73	53.00	**58.91**	25.49
Certainty-Rare	54.73	42.27	45.23	**47.30**
Uncertainty-Common	54.73	51.57	47.20	44.03
Uncertainty-Rare	54.73	46.00	38.13	21.65
Submodular Selection (Diversity-Aware)				
Certainty (Sub)	54.73	53.88	**62.35**	43.89
Uncertainty (Sub)	54.73	50.11	55.55	45.23
Certainty-Common (Sub)	54.73	**55.59**	60.22	42.33
Certainty-Rare (Sub)	54.73	47.59	53.59	**50.97**
Uncertainty-Common (Sub)	54.73	53.17	55.47	45.21
Uncertainty-Rare (Sub)	54.73	49.48	47.27	35.30

Table 9: Lemmatization Performance on Different Continual Learning Methods at TB $(10,000)$ - Curras (Palestinian). Note:5,7,14,21 are Active Learning Iteration.

Method	It.5	It.7	It.14	It.21
Baselines				
without Continual Learning	97.43	97.45	97.38	97.28
Experience Replay	97.43	78.09	79.98	77.85
Heuristic Selection (Top-N Sorting)				
Common Freq.	97.43	96.47	92.30	70.12
Rare Freq.	97.43	96.61	**97.22**	94.67
Max BT (Certainty)	97.43	97.11	91.35	80.97
Min BT (Uncertainty)	97.43	97.06	96.61	**95.54**
Certainty-Common	97.43	97.16	91.53	70.09
Certainty-Rare	97.43	**97.21**	92.70	91.74
Uncertainty-Common	97.43	97.06	97.09	94.19
Uncertainty-Rare	97.43	97.09	97.36	95.43
Submodular Selection (Diversity-Aware)				
Certainty (Sub)	97.43	**97.43**	93	91.30
Uncertainty (Sub)	97.43	97.19	**97.58**	**96.38**
Certainty-Common (Sub)	97.43	97.26	93.10	90.25
Certainty-Rare (Sub)	97.43	97.37	93.53	93.40
Uncertainty-Common (Sub)	97.43	97.15	97.49	96.07
Uncertainty-Rare (Sub)	97.43	97.13	97.53	96.26

Table 10: Prefix Tagging Performance on Different Continual Learning Methods at TB $(10,000)$ - Curras (Palestinian). Note:5,7,14,21 are Active Learning Iteration.

Method	It.5	It.7	It.14	It.21
Baselines				
without Continual Learning	92.95	93.30	94.21	94.13
Experience Replay	92.95	76.59	77.60	77.35
Heuristic Selection (Top-N Sorting)				
Common Freq.	92.95	91.64	87.75	73.15
Rare Freq.	92.95	90.39	89.96	**88.88**
Max BT (Certainty)	92.95	91.97	90.29	74.53
Min BT (Uncertainty)	92.95	91.86	89.97	84.80
Certainty-Common	92.95	92.20	90.00	72.36
Certainty-Rare	92.95	92.29	**91.63**	86.42
Uncertainty-Common	92.95	91.79	89.41	83.81
Uncertainty-Rare	92.95	**92.62**	89.48	85.67
Submodular Selection (Diversity-Aware)				
Certainty (Sub)	92.95	92.24	90.31	85.09
Uncertainty (Sub)	92.95	91.76	90.94	85.61
Certainty-Common (Sub)	92.95	92.12	90.06	84.51
Certainty-Rare (Sub)	92.95	**92.43**	**91.56**	**88.12**
Uncertainty-Common (Sub)	92.95	92.00	90.88	86.16
Uncertainty-Rare (Sub)	92.95	92.30	90.76	86.35

Table 11: Suffix Tagging Performance on Different Continual Learning Methods at TB $(10,000)$ - Curras (Palestinian). Note:5,7,14,21 are Active Learning Iteration.

D Continual Learning Results at budget 20,000

Table 12 reports the performance of the best-performing heuristic and submodular continual methods compared with both baselines at the final active learning iteration (21) across all tasks at training budget $20,000$.

Tables 13, 14, 15, and 16 report the performance of POS tagging, lemmatization, prefix tagging, and suffix tagging, respectively, across key active learning iteration milestones (10, 13, 17, and 21) under a fixed training budget (TB) of 20,000 tokens for each continual learning method. Notable here Iteration 10 serves as the anchor point, where the model reaches the maximum number of tokens defined by the TB where here is $20,000$. From Iteration 11 onward, continual learning mechanisms are activated to manage history preservation while active

Task	Baselines		Top Heuristic Selection		Top Submodular Selection	
	Without Continual	Experience Replay	Strategy	F_1	Strategy	F_1
POS tagging	95.18	79.89	Uncertainty-Common	91.42	Uncertainty-Rare (Sub)	**92.57**
Lemmatization	75.50	62.77	Certainty-Rare	58.94	Certainty-Rare (Sub)	**58.94**
Prefix Tagging	97.28	82.49	Uncertainty-Common	96.72	Uncertainty-Common (Sub)	**96.73**
Suffix Tagging	94.13	79.88	Rare Freq.	**92.33**	Certainty-Rare (Sub)	90.29

Table 12: Performance of the best-performing heuristic and submodular for each task at the final active learning iteration (21) under training budget $TB = 20,000$ on Curras (Palestinian).

learning continues to query new data sequentially.

Method	It.10	It.13	It.17	It.21
Baselines				
without Continual Learning	95.03	95.19	94.87	95.18
Experience Replay	95.03	81.81	81.37	79.89
Heuristic Selection (Top-N Sorting)				
Common Freq.	95.03	94.33	92.41	84.46
Rare Freq.	95.03	90.69	82.80	84.63
Max BT (Certainty)	95.03	94.33	89.89	86.88
Min BT (Uncertainty)	95.03	**94.47**	84.41	90.51
Certainty-Common	95.03	94.04	89.98	86.51
Certainty-Rare	95.03	94.12	91.19	90.80
Uncertainty-Common	95.03	92.51	**94.77**	**91.42**
Uncertainty-Rare	95.03	91.63	84.02	89.52
Submodular Selection (Diversity-Aware)				
Certainty (Sub)	95.03	93.85	90.98	89.97
Uncertainty (Sub)	95.03	93.21	92.92	91.58
Certainty-Common (Sub)	95.03	93.88	90.36	88.76
Certainty-Rare (Sub)	95.03	**94.64**	91.30	91.19
Uncertainty-Common (Sub)	95.03	92.88	**94.59**	91.36
Uncertainty-Rare (Sub)	95.03	93.59	92.91	**92.57**

Table 13: POS Performance on Different Continual Learning Methods at TB $(20,000)$ - Curras (Palestinian). Note: 10, 13, 17, 21 are Active Learning Iterations.

Method	It.10	It.13	It.17	It.21
Baselines				
without Continual Learning	66.63	73.62	74.53	75.50
Experience Replay	66.63	65.36	65.11	62.77
Heuristic Selection (Top-N Sorting)				
Common Freq.	66.63	72.10	67.02	48.74
Rare Freq.	66.63	52.56	49.05	48.63
Max BT (Certainty)	66.63	73.08	**65.49**	48.92
Min BT (Uncertainty)	66.63	62.02	49.47	48.05
Certainty-Common	66.63	**72.84**	64.98	49.54
Certainty-Rare	66.63	72.02	58.28	**58.94**
Uncertainty-Common	66.63	60.56	57.58	45.22
Uncertainty-Rare	66.63	59.80	48.24	39.88
Submodular Selection (Diversity-Aware)				
Certainty (Sub)	66.63	**73.26**	**66.04**	51.57
Uncertainty (Sub)	66.63	66.90	53.55	46.19
Certainty-Common (Sub)	66.63	72.59	65.74	50.73
Certainty-Rare (Sub)	66.63	72.20	65.08	**58.94**
Uncertainty-Common (Sub)	66.63	65.53	60.02	50.09
Uncertainty-Rare (Sub)	66.63	68.21	54.22	45.25

Table 14: Lemmatization Performance on Different Continual Learning Methods at TB $(20,000)$ - Curras (Palestinian). Note: 10, 13, 17, 21 are Active Learning Iterations.

Complementing the performance matrix, Table 17 presents the average iteration time incurred by each method,which is the average replay selection time and the average training time at TB $20,000$.

It is observed from Table 12 and the detailed task-specific tables (13, 14, 15, and 16) that the same

Method	It.10	It.13	It.17	It.21
Baselines				
without Continual Learning	97.64	97.27	97.40	97.28
Experience Replay	97.64	84.66	83.68	82.49
Heuristic Selection (Top-N Sorting)				
Common Freq.	97.64	96.59	93.75	88.65
Rare Freq.	97.64	96.40	**96.90**	95.57
Max BT (Certainty)	97.64	**97.57**	93.78	91.35
Min BT (Uncertainty)	97.64	97.12	96.81	96.65
Certainty-Common	97.64	96.99	94.86	91.93
Certainty-Rare	97.64	97.18	96.55	94.85
Uncertainty-Common	97.64	96.94	96.52	**96.72**
Uncertainty-Rare	97.64	96.59	96.24	96.63
Submodular Selection (Diversity-Aware)				
Certainty (Sub)	97.64	96.93	95.21	92.85
Uncertainty (Sub)	97.64	97.04	96.78	96.50
Certainty-Common (Sub)	97.64	96.97	94.88	92.68
Certainty-Rare (Sub)	97.64	97.07	96.06	94.93
Uncertainty-Common (Sub)	97.64	97.14	96.37	**96.73**
Uncertainty-Rare (Sub)	97.64	**97.43**	**96.99**	96.39

Table 15: Prefix Tagging Performance on Different Continual Learning Methods at TB $(20,000)$ - Curras (Palestinian). Note: 10, 13, 17, 21 are Active Learning Iterations.

Method	It.10	It.13	It.17	It.21
Baselines				
without Continual Learning	94.16	94.16	94.14	94.13
Experience Replay	94.16	81.05	80.75	79.88
Heuristic Selection (Top-N Sorting)				
Common Freq.	94.16	93.12	90.25	83.72
Rare Freq.	94.16	93.45	92.73	**92.33**
Max BT (Certainty)	94.16	93.92	91.77	86.78
Min BT (Uncertainty)	94.16	92.90	93.40	88.50
Certainty-Common	94.16	93.80	91.72	87.21
Certainty-Rare	94.16	**93.92**	93.04	89.83
Uncertainty-Common	94.16	91.96	93.30	88.27
Uncertainty-Rare	94.16	92.63	**93.85**	88.41
Submodular Selection (Diversity-Aware)				
Certainty (Sub)	94.16	93.70	91.88	88.44
Uncertainty (Sub)	94.16	92.81	93.40	88.38
Certainty-Common (Sub)	94.16	93.72	91.64	88.26
Certainty-Rare (Sub)	94.16	**93.85**	93.22	**90.29**
Uncertainty-Common (Sub)	94.16	91.85	93.03	88.50
Uncertainty-Rare (Sub)	94.16	93.13	**93.61**	88.68

Table 16: Suffix Tagging Performance on Different Continual Learning Methods at TB $(20,000)$ - Curras (Palestinian). Note: 10, 13, 17, 21 are Active Learning Iterations.

performance trends are maintained across methods. POS and prefix tagging achieve the best results with uncertainty-based methods, while lemmatization performs best with certainty-based methods. In contrast, suffix tagging is influenced mainly by the rarity of morphological patterns. A similar observation is also noted for re-finetuning time preservation, where *Uncertainty-Common* consistently

Method	Selection Time	Training Time	Total Time
Baselines			
without Continual Learning	00.00	13.79	13.79
Experience Replay	0.001	13.29	13.29
Heuristic Selection (Top-N Sorting)			
Common Freq.	00.10	13.02	13.12
Rare Freq.	00.10	13.31	13.41
Max BT (Certainty)	00.08	13.13	13.21
Min BT (Uncertainty)	00.07	13.26	13.33
Certainty-Common	00.10	13.00	13.10
Certainty-Rare	00.09	13.29	13.39
Uncertainty-Common	00.09	11.63	11.72
Uncertainty-Rare	00.09	13.16	13.25
Submodular Selection (Diversity-Aware)			
Certainty (Sub)	1.29	12.91	14.20
Uncertainty (Sub)	1.30	12.82	14.12
Certainty-Common (Sub)	1.36	13.84	15.20
Certainty-Rare (Sub)	1.36	13.43	14.79
Uncertainty-Common (Sub)	1.36	13.35	14.71
Uncertainty-Rare (Sub)	1.34	13.38	14.72

Table 17: Replay Selection and Training Time (Minutes) on Different Continual Learning Methods at Training Budget (TB) $20,000$.

saves approximately 2.1 minutes per iteration.