MetAL: A Novel Meta Active Learning Approach for Morphophonological Processing

Anonymous ACL submission

Abstract

In morphologically complex languages like Arabic, developing a morphophonological processing system poses significant challenges. While deep learning models have shown success in this task, these models heavily rely on the size of the annotated data. However, creating large datasets, especially for low-resource languages such as different Arabic dialects, is very time-consuming, hard, and expensive. Furthermore, not all annotated data contribute beneficial information for training models. To address these issues, active learning tries to guide the learning algorithm to choose informative samples for annotation. Despite the limited research on applying active learning to morphophonological processing, this paper introduces a novel combination of meta and active learning approaches for tackling this task. To the best of our knowledge, there is no research that focuses on the combination of these approaches. The experimental results conducted on Egyptian Arabic demonstrate that achieving similar performance as the state-of-the-art model on the entire dataset is possible with only approximately 23% of annotated data. Notably, our proposed method outperforms existing successful deep active learning methods.

1 Introduction

During the last few years, morphological inflection processing (Narasimhan et al., 2015; Kirov and Cotterell, 2018; Belth et al., 2021) has received a great deal of attention. This task has gained significant attention in the NLP community (Halabi, 2016; Khalifa et al., 2020; Alhafni et al., 2020) and has recently been the focus of several shared tasks (Cotterell et al., 2018; Vylomova et al., 2020; Batsuren et al., 2022). Neural seq2seq models achieved impressive precision, particularly when accessing extensive training datasets (Cotterell et al., 2018). Nonetheless, when trained with limited data, these neural models exhibit low performance, often for languages with complex morphological structures. However, having enough annotated data is challenging, costly, and time-intensive. In addition, all annotated data do not contain useful information for enhancing the quality of the learning algorithm.

In this paper, we have introduced a novel meta active learning (MetAL) approach to reduce the amount of labeled data required for the Egyptian Arabic morphophonological processing. Morphophonological processing takes a sequence of morphs and applies morphophonological processes to obtain the surface form. It is an important component of inflection. In our experiments, we have selected an efficient transformer model for character-level transduction tasks (Wu et al., 2021) as our baseline model. We have used the pool-based active learning method with maximum entropy criterion for selecting uncertain samples. By combining meta and active learning methods, we have achieved similar results as supervised learning with only 23% of the training dataset, which is a SOTA result in this area. Our approach also surpasses currently effective deep active learning (DAL) techniques, especially in scenarios where a small amount of annotated data is involved. To the best of our knowledge, our work is the first application of a MetAL approach in the morphophonological processing task. It should be noted that our proposed method is not specific to Arabic, and it can be also applied to other languages.

2 Previous Work

There has been extensive non-neural network research focused on Arabic morphological modeling, which includes morphophonological processing (Habash and Rambow, 2006; Graff et al., 2009; Habash et al., 2022). Recently, with the popularity of neural networks, various studies of DL models based on character-level neural transducers using transformers and RNNs approaches have been introduced (Wu et al., 2021; Dankers et al., 2021; Yang et al., 2022; Wehrli et al., 2022).

DAL has recently been used in various sub-fields of NLP such as NER (Prabhu et al., 2019; Liu et al., 2020) and machine translation (Liu et al., 2018; Zhao et al., 2020). In the morphological inflection task, Muradoglu and Hulden (2022) presented a novel DAL technique using word-level entropy for lemma inflection in different languages. Mirbostani et al. (2023) introduced another DAL approach that utilizes character-level entropy. This method achieved superior performance compared to previous techniques.

In recent years, meta-learning has found successful applications in various NLP domains such as machine translation (Gu et al., 2018), NER (Ma et al., 2022), and semantic parsing (Langedijk et al., 2022). It aims to address the challenge of quickly adapting to new training data. Common meta-learning techniques can be classified into three groups: black-box adaptation (Santoro et al., 2016), optimization (Finn et al., 2017), and metric learning (Snell et al., 2017). Kann et al. (2020) introduced a meta-learning-based approach for cross-lingual transfer learning in the morphological inflection task using the MAML algorithm (Finn et al., 2017).

3 Problem Definition and Dataset

In this paper, we focus on morphophonological processing, in which a model receives an underlying representation (UR) of a word and generate its surface form (SF), i.e., spoken form. We use the morphophonology dataset of Khalifa et al. (2022), which consists of (UR, SF) pairs for Egyptian Arabic and is derived from the ECAL dataset (Kilany et al., 2002). The UR includes segmentation information, using $\#$ to indicate word boundaries, $-$ for prefixes, and $=$ for suffixes. The dataset's split into TRAIN, DEV, and EVAL is based on the ECAL's split. Since in ECAL, the split is based on running texts, some words may appear in multiple splits. To address this, Khalifa et al. (2022) also created DEV-OOV and EVAL-OOV subsets by including only words that do not overlap with TRAIN. Some samples of (UR, SF) pairs and the dataset's statistics are shown in appendix section.

4 Proposed Method

4.1 Baseline Network

We performed various experiments to choose the best baseline model for our MetAL experiments, and chose Wu et al. (2021)'s transformer-based model. It has outperformed existing RNN-based seq2seq models and achieved SOTA results on our target dataset.

4.2 Meta Learning Method

Given the limited resources of character-level datasets for morphophonological processing tasks, an optimization-based meta-learning algorithm is a practical method to achieve high accuracies with small datasets and few training iterations. Accordingly, we have adopted MAML (Finn et al., 2017), a general optimization algorithm compatible with gradient descent, in our method. The primary assumption is that our neural transducer model, f_θ, is parameterized by θ, and its loss function, $\mathcal{L}$, is minimized using a gradient-based learning technique. The model is trained over multiple tasks, $\mathcal{T}_i$, sampled from a distribution over tasks $p(\mathcal{T})$ to which the model should be adapted.

In our problem, each task has an associated dataset containing pairs of (UR, SF) for words. It is split into two subsets: *support* and *query*. The support set (i.e., the training set) is a labeled subset used for adaptation of the meta-learning model through learning the initial parameters. The query set (i.e., the test set) is the complement of the support set used for evaluating the model's performance on new, unseen data points. Furthermore, it indicates how well the model generalizes over the tasks not seen in the meta-training phase.

Model adaptation is the process of quickly learning and adjusting the initial parameters to a task using a small number of samples. Equation (1) computes adapted parameters of the model, f_θ, over a single task, $\mathcal{T}_i$, using one adaptation iteration (i.e., $N_{\text{adapt}} = 1$).

$$\theta_i^{'} = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \quad (1)$$

In the given equation, α is the step size, which can be a fixed or a learned hyperparameter.

Applying multiple gradient updates on the meta-learning model creates a trade-off between parameter refinement and adaptation speed. Therefore, choosing a suitable N_{adapt} depends on the complexity of the task, as more resources and training time is required to reach a thorough adaptation.

The optimal parameters of the model for better performance of $f_{\theta_i^{'}}$ in terms of θ, adapted over multiple tasks sampled from $p(\mathcal{T})$, are computed using Equation (2).

$$\theta^* = \arg\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \quad (2)$$

This method significantly improves the fine-tuning of the model, with one or more adaptation iterations (i.e., $N_{\text{adapt}} \geq 1$) on a new task from the same distribution.

Given the adapted parameters over each task, θ_i', the optimization of the model across all the sampled tasks, $\mathcal{T}_i$, is performed by updating the model parameters, θ, using Equation (3):

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \quad (3)$$

where β is the meta step size.

4.3 Meta Active Learning (MetAL) Method

In our approach, we have combined the principles of meta-learning, described in Section 4.2, and active learning (AL) to improve the accuracy of the baseline model on our problem. The meta-learning model shares the knowledge learned from previous tasks of the same distribution with the new tasks that are generated based on the most informative instances in each AL training cycle. In our algorithm, we have used the pool-based AL method integrated with the maximum entropy criterion for selecting uncertain samples in each training cycle.

The MetAL method initiates the training procedure by random sampling, without replacement, of a portion (around 10%) of a pool dataset, $\mathcal{U}$. In our case, $\mathcal{U}$ initially contains 13,170 unannotated samples. In the next step, these randomly selected samples are annotated and divided into two subsets: *tuning* and *training*. A tuning subset, $\mathcal{V}$, is a dataset (500 samples in our experiments) used for validating the trained models. A training subset, $\mathcal{D}$, is an augmented dataset (900 samples in our experiments) used for training the model in the initial cycle according to the meta-learning method. In subsequent training cycles, $\mathcal{V}$ remains constant; however, $\mathcal{D}$ is increased by δ (=250) number of samples selected from the remaining samples of $\mathcal{U}$ based on their maximum entropy values.

For a given sample from $\mathcal{U}$, Equation (4) computes the entropy value of a word, w, in terms of its characters' logits, $\mathbf{c}$, predicted by the most performant model trained in the previous AL cycle.

$$H(w) = \max_{\mathbf{c} \in w}(-\sum_{i=1}^{N} p_i(\mathbf{c}) \log p_i(\mathbf{c})) \quad (4)$$

$p_i(\mathbf{c})$ is the probability value of the i^{th} character in an N-sized $\mathbf{c}$ and is calculated using the softmax function, $e^{c_i}/\sum_{i=1}^{N} e^{c_i}$. Given that the model's predicted characters with the lowest confidence value have the highest entropy, the entropy of a word, $H(w)$, corresponds to the maximum value of all the entropy values associated with its respective characters.

A MetAL training cycle encompasses a meta-learning method with $N_{\mathcal{T}}$ number of tasks. The training dataset, $\mathcal{D}$, is randomly divided into $N_{\mathcal{T}}$ equal subsets, $\mathcal{D}_i$. Each task, $\mathcal{T}_i$, has an associated $\mathcal{D}_i$, and during meta-learning iterations, undertakes a meta-training and a meta-testing phase. The $\mathcal{D}_i$ is split into a support set, $\mathcal{S}_i$, and a query set, $\mathcal{Q}_i$, to be used in those phases, respectively.

K_s sample batches are selected sequentially from $\mathcal{S}_i$ for meta-learning model adaptation during meta-training. This phase helps the model acquire knowledge and adaptability across current tasks and learn to generalize for the next tasks by updating the meta-learning model's parameters using Equations (1) and (2). The meta-testing phase involves exposing the model to the query set for parameters optimization across all tasks, given the adapted parameters over each task. K_q shots of sample batches are randomly chosen from $\mathcal{Q}_i$ for fine-tuning the model using Equation (3).

5 Experimental Results

We evaluated our proposed MetAL method by training the baseline model with MetAL, DAL, and random training (i.e., passive learning) over 5 runs, with each run utilizing a randomly chosen tuning set based on different seed values, and evaluating on EVAL, EVAL-OOV, DEV, and DEV-OOV datasets. Figure 1 shows the mean and standard deviation (SD) of the model's performance on the morphophonological processing task in terms of accuracy for each training cycle. It reflects the average outcome across all 5 runs. During our analysis, no significant loss discrepancy between the training and tuning sets has been observed, and both the mean and SD of the results display relatively minimal variations. The details regarding the hyper-parameters and variables employed in our experiments and the supplementary experiments are presented in Appendices B and C.

As shown in Figure 1, the curve corresponding to our proposed MetAL method displays a shape that tends towards an asymptote. It demonstrates a

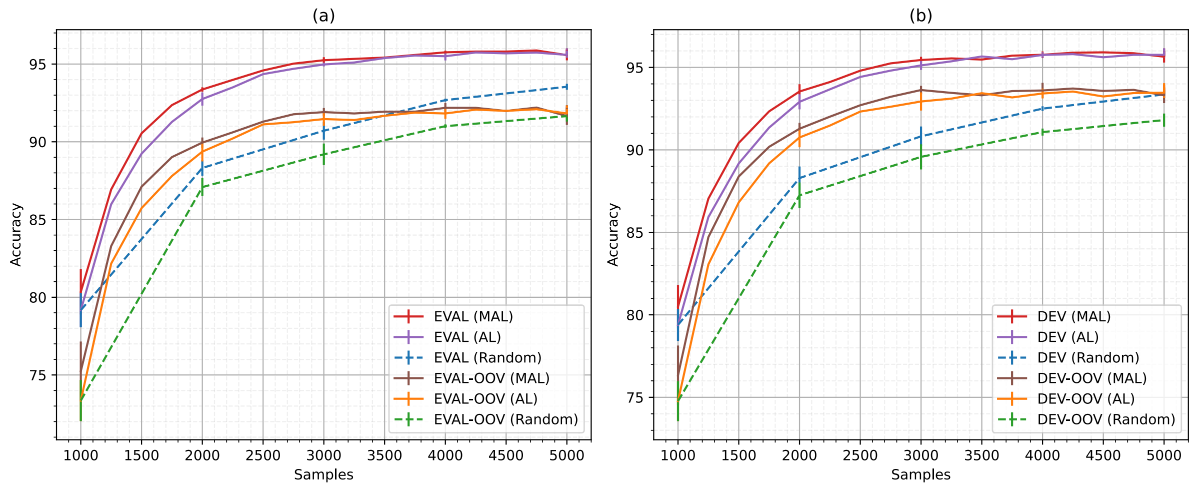

Figure 1: Morphophonological processing task: The mean and standard deviation of the baseline model's accuracy trained using MetAL, DAL, and random training methods, and evaluated on (a) EVAL, EVAL-OOV, (b) DEV, and DEV-OOV datasets

considerably accelerated growth in contrast to the DAL and random curves and initiates with a higher level of accuracy, outperforming both alternative training methods.

By achieving optimal accuracy with only approximately 3,000 samples (i.e., about 23% of the training set) on EVAL-OOV and DEV-OOV, and only around 4,000 samples (i.e., about 30% of the training set) on EVAL and DEV, our method demonstrates its capacity to actively elicit a small set of informative samples from the pool for labeling and effectively adapt to these samples with few shots, showing strong performance on out-of-vocabulary datasets of the same distribution. Hence, it reveals that this method excels over DAL on small-scale datasets. In contrast, the random learner relies on whole training set to reach its best accuracy level.

The SD of our MetAL method (over the experimental results using 5 runs) has a consistent decrease in each subsequent training cycle, in addition to being lower compared to that of the DAL and random training methods. This can be attributed to the effective utilization of data enabled by the rapid adaptation to new tasks in meta-learning, increasing the model's robustness, reducing the impact of variations in the training set, and resulting in a lower SD. In contrast, DAL focuses on selecting informative instances, which may not directly address data variability.

Adopting the meta-learning method to go over multiple tasks during each training iteration, the model has developed the capacity to generalize its gained experience from sampled tasks to improve over unseen samples of the same distribution. In our approach, exposing the model to a few shots of informative data points extracted from a small-sized AL training cycle dataset, $\mathcal{D}$, greatly reduces the required annotation and accelerates the model's training process compared to employing DAL method alone.

6 Conclusion

We have introduced the meta active learning algorithm, a combination of meta and active learning approaches, using the morphophonological processing task for Egyptian Arabic dialect as a sample task. The results of our experiment demonstrate that achieving similar accuracy as the SOTA model on the entire dataset is possible with only 23% of the total training dataset, which outperforms existing successful deep active learning methods, especially on lower amounts of annotated data.

Our method has been designed to be language and model agnostic. We hypothesize that by concentrating on Arabic, a language renowned for its morphological intricacies, our approach's efficacy will extend to diverse languages. As our prospective research topics, we suggest addressing the intricacies of templatic morphology, a substantial source of complexity within Arabic, in addition to analyzing the application of our method to train other baseline models, generate datasets for low-resource Arabic dialects and other languages, and incorporate alternative uncertainty criteria.

Limitations

Our work, like many deep learning algorithms, relies on GPU resources. In common learning problems, models are trained once on training datasets, tuned on the development sets, and then ready for inference. However, the training process involves conducting multiple iterations whenever new informative samples from the pool are annotated and added to the training set throughout AL cycles. As the augmented training set grows, the demand for GPU resources increases.

Furthermore, there is a trade-off between the adaptation speed and the generalization performance during the meta-learning phase. Additional adaptation iterations and support shots are required for broader task generalization in meta-learning, increasing GPU resource demand. However, the requirement for GPU resources is not specifically tied to our proposed method but rather stems from the inherent nature of active learning and meta-learning methods.

Our algorithm is language and model agnostic; however, it has only been evaluated on the Egyptian Arabic dialect. Therefore, further research is needed to examine the accuracy of the model across other languages and dialects using different learning models.

It is worth mentioning that the proposed method exhibits the potential to achieve higher accuracy with increased hyperparameter values. Unfortunately, due to hardware limitations, we were unable to perform the experiments to validate this.

Ethics Statement

This study is primarily focused on fundamental research and is not related to a specific application. We do not anticipate any ethical concerns arising from the algorithms and technologies proposed in this work. This research has utilized datasets and open-source libraries that have been previously published and publicly accessible.

References

Bashar Alhafni, Nizar Habash, and Houda Bouamor. 2020. Gender-aware reinflection using linguistically enhanced neural models. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 139–150.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Caleb Belth, Sarah Payne, Deniz Beser, Jordan Kodner, and Charles Yang. 2021. The greedy and recursive search for morphological productivity. *arXiv preprint arXiv:2105.05790*.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels. Association for Computational Linguistics.

Verna Dankers, Anna Langedijk, Kate McCurdy, Adina Williams, and Dieuwke Hupkes. 2021. Generalising to German plural noun classes, from the perspective of a recurrent neural network. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 94–108.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Nizar Habash, Reham Marzouk, Christian Khairallah, and Salam Khalifa. 2022. Morphotactic modeling in an open-source multi-dialectal Arabic morphological analyzer and generator. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 92–102, Seattle, Washington. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2006. MAGEAD: A morphological analyzer and generator for the Arabic dialects. In *Proceedings of the International Conference on Computational Linguistics and the Conference of the Association for Computational Linguistics (COLING-ACL)*, pages 681–688, Sydney, Australia.

Nawar Halabi. 2016. *Modern standard Arabic phonetics for speech synthesis*. Ph.D. thesis, University of Southampton.

Katharina Kann, Samuel R Bowman, and Kyunghyun Cho. 2020. Learning to learn morphological inflection for resource-poor languages. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8058–8065.

Salam Khalifa, Jordan Kodner, and Owen Rambow. 2022. Towards learning arabic morphophonology. In *Proceedings of the seventh Arabic Natural Language Processing Workshop (WANLP) at EMNLP 2022*, pages 295–301s.

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 12th Language Resources and Evaluation Conference*, pages 3895–3904.

H Kilany, H Gadalla, H Arram, A Yacoub, A El-Habashi, and C McLemore. 2002. Egyptian colloquial arabic lexicon. *LDC catalog number LDC99L22*.

Christo Kirov and Ryan Cotterell. 2018. Recurrent neural networks in linguistic theory: Revisiting pinker and prince (1988) and the past tense debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Anna Langedijk, Verna Dankers, Phillip Lippe, Sander Bos, Bryan Cardenas Guevara, Helen Yannakoudakis, and Ekaterina Shutova. 2022. Meta-learning for fast cross-lingual adaptation in dependency parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8503–8520, Dublin, Ireland. Association for Computational Linguistics.

Ming Liu, Wray Buntine, and Gholamreza Haffari. 2018. Learning to actively learn neural machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 334–344.

Mingyi Liu, Zhiying Tu, Zhongjie Wang, and Xiaofei Xu. 2020. Ltp: A new active learning strategy for bert-crf based named entity recognition. *ArXiv*, abs/2001.02524.

Tingting Ma, Huiqiang Jiang, Qianhui Wu, Tiejun Zhao, and Chin-Yew Lin. 2022. Decomposed meta-learning for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1584–1596, Dublin, Ireland. Association for Computational Linguistics.

Seyed Morteza Mirbostani, Yasaman Boreshban, Salam Khalifa, SeyedAbolghasem Mirroshandel, and Owen Rambow. 2023. Deep active learning for morphophonological processing. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Toronto, Canada. Association for Computational Linguistics.

Saliha Muradoglu and Mans Hulden. 2022. Eeny, meeny, miny, moe. how to choose data for morphological inflection. *arXiv preprint arXiv:2210.14465*.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167.

Ameya Prabhu, Charles Dognin, and Maneesh Singh. 2019. Sampling bias in deep active classification: An empirical study. *arXiv preprint arXiv:1909.09389*.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics.

Silvan Wehrli, Simon Clematide, and Peter Makarov. 2022. Cluzh at sigmorphon 2022 shared tasks on morpheme segmentation and inflection generation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 212–219.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.

Changbing Yang, Garrett Nicolai, Miikka Silfverberg, et al. 2022. Generalizing morphological inflection systems to unseen lemmas. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 226–235.

Yuekai Zhao, Haoran Zhang, Shuchang Zhou, and Zhihua Zhang. 2020. Active learning approaches to enhancing neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1796–1806.

A Dataset

For the purpose of this paper, our concentration lies on the intricate Arabic morphophonological processing task, given its substantial complexity and variation among the dialects. Accordingly, we have used the annotated Egyptian Arabic morphophonology dataset introduced by Khalifa et al. (2020) to assess our proposed method on a low-resource language with a high degree of morphological complexity. The dataset contains pairs of (UR, SF) splitted into TRAIN, DEV, and EVAL subsets, based on the ECAL's splits introduced by Kilany et al. (2002), in addition to EVAL-OOV and DEV-OOV subsets, specific to this dataset. Two samples of (UR, SF) pairs along with the sizes of different splits of the dataset used in our task are shown in Tables A.1 and A.2.

UR	SF
#qAl=li=hum#	#qalluhum#
#bi-t-Akl=I#	#bitakli#

Table A.1: Two samples of (UR, SF) pairs of the dataset.

	TRAIN	DEV	EVAL
All	13,170	5,180	6,974
OOV	-	2,189	2,271

Table A.2: The sizes of different splits of the dataset.

Owing to the dataset's comprehensive annotations, we undertook our experiments using a simulation-based active learning approach. As outlined in Section 4.3, during each MetAL cycle, we deliberately select K_s samples as the support set from the pool of annotated training samples, sequentially presenting it to the model. Conversely, the query set, containing K_q samples, is formed from the complementary set of the support set and is introduced to the model in a randomized manner.

B Experimental Setup

We performed a series of experiments involving various successful approaches, including the Neural Transducer by Wu et al. (2021), and Cluzh by Wehrli et al. (2022). As a result, we selected the character-level neural transducer (i.e., Wu et al. (2021) system) as our baseline model. This choice stems from its standing as a SOTA transformer-based model, outperforming existing RNN-based seq2seq models and showcasing successful outcomes across the entirety of the Arabic morphophonological processing dataset. The model is a compact transformer consisting of 4 encoder-decoder layers, 4 self-attention heads, an embedding dimension of 256, and a hidden size of 1024 for the feed-forward layer. The model has 7.37M parameters, excluding embeddings and the pre-softmax linear layer.

The optimal values for the hyper-parameters of our experiments are listed in Table B.1. We conducted multiple experiments with different values to analyze the performance of our proposed MetAL method. The mean and standard deviation of the results in terms of accuracy for each training cycle is reported in Figure 1.

Parameter	**Value**
$\mathcal{D}$ (initial) samples	900
$\mathcal{V}$ samples	500
δ samples	250
Support Feeding Methodology	rotation
Query Feeding Methodology	random
Uncertainty criterion	entropy
Training batch size (BS)	100
Evaluation batch size	6
α learning rate	0.001
β learning rate	0.0001
Dropout	0.3
$N_{\mathcal{T}}$	4
K_s	8
K_q	8
N_{adapt}	1

Table B.1: The optimal hyper-parameter values of the experiments.

We have used PyTorch, NumPy, Pandas, and Matplotlib software packages to implement the proposed algorithm. The experiments were performed on a hardware comprising an Intel Core i7-8700K CPU with 6 cores running at 3.70GHz speed, a GeForce GTX 1080 GPU with 8GB of VRAM, and 64GB of RAM. Each MetAL training cycle needs a minimum of 7.92GB of GPU memory and 4.86GB of RAM.

C Supplementary Experiments

In addition to showcasing the most optimal hyper-parameter values in the paper, as shown in Table B.1, we conducted a thorough study to illustrate the importance of the hyper-parameters and components of our proposed MetAL approach. We tuned essential hyper-parameters such as the number of tasks (N_T), support shots (K_s), query shots (K_q), adaptation iterations (N_{adapt}), training batch size (BS), and feeding methodology for passing batched samples to the model (FM). For instance, the examples demonstrated in Table C.1 outline our model's performance on the tune set. We conducted five experiments in which we employed different seed values to randomly select tuning sets, and the results represent the average outcome derived from all five experiments.

$N_{\mathcal{T}}$	K_s	K_q	N_{adapt}	BS	Support FM	Query FM	Epochs	Accuracy
1	1	1	1	100	random	random	667	44.08%
4	1	1	1	100	random	random	559	89.45%
4	2	2	1	100	random	random	650	91.90%
4	6	2	1	100	random	random	980	93.02%
4	6	6	1	100	random	random	680	94.71%
4	6	6	1	400	random	random	700	95.14%
4	6	6	1	100	rotation	random	760	95.37%

Table C.1: The effects of different hyperparameters on the accuracy of the model on the tune set.