Uncertainty-Driven Partial Diacritization for Arabic Text

Humaid Alblooshi, Artem Shelmanov, Hanan Aldarmaki
Department of Natural Language Processing
Mohamed Bin Zayed University of Artificial Intelligence, UAE
{Humaid.alblooshi, Artem.Shelmanov, Hanan.Aldarmaki}@mbzuai.ac.ae

Abstract

We propose an uncertainty-based approach to Partial Diacritization (PD) for Arabic text. We evaluate three uncertainty metrics for this task: Softmax Response, BALD via MC-dropout, and Mahalanobis Distance. We further introduce a lightweight Confident Error Regularizer to improve model calibration. Our preliminary exploration illustrates possible ways to use uncertainty estimation for selectively retaining or discarding diacritics in Arabic text with an analysis of performance in terms of correlation with diacritic error rates. For instance, the model can be used to detect words with high diacritic error rates which tend to have higher uncertainty scores at inference time. On the Tashkeela dataset, the method maintains low diacritic error rates while reducing the amount of visible diacritics on the text by up to 50% with thresholding-based retention.

1 Introduction

Arabic script relies on diacritics, commonly referred to in Arabic as Tashkeel (تَشْكِيل) to mark short vowels, gemination, and other phonemic distinctions that may not be represented by the base letters. Fully-diacritized text eliminates ambiguity and supports precise pronunciation, which is helpful for applications such as text-to-speech (TTS) synthesis, machine translation, and language learning (Mubarak et al., 2019; Lameris, 2021). However, when every letter carries its diacritic, the resulting text becomes visually dense and can slow down readers (ElNokrashy and AlKhamissi, 2024; Roman and Pavard, 1987). Partial diacritization can be employed to balance disambiguation and readability and to optimize performance in downstream NLP applications.

State-of-the-art transformer-based diacritization models can achieve diacritic error rates (DER) below 2% on standard benchmarks (Assad et al., 2024), but their performance may degrade in out-of-domain data (Toyin et al., 2025a). These models often operate as "black boxes," where model outputs are accepted regardless of confidence scores. Their outputs are often fully diacritized, which increases visual complexity and can slow down reading speed and reduce clarity. Furthermore, many diacritics are redundant, particularly in common words where pronunciation is intuitive or easily inferred. These factors make Fully Diacritized (FD) text less practical for general application, motivating the need for Partially Diacritized (PD) text in such settings. Prior studies proposed computational approaches that rely on heuristics and morphological analysis to perform partial diacritization (Diab et al., 2007; Alqahtani et al., 2019). Others have proposed neural networks (Fadel et al., 2019) with some success. However, research on partial diacritization remains limited, largely due to the difficulty of evaluating performance; optimal partial diacritization is an illusive concept with no standard evaluation framework or metrics.

In this paper, we explore an uncertainty-driven framework for PD and provide a preliminary intrinsic evaluation of this framework through error analysis. We evaluate three uncertainty metrics: Softmax Response, Bayesian Active Learning by Disagreement (BALD) via Monte Carlo dropout, and Mahalanobis distance in latent feature space. At inference time, the predicted diacritic of each character is compared with a chosen threshold θ. We may keep the diacritic if the uncertainty score is above or below said threshold, allowing for flexibility in application. To mitigate the well-known overconfidence of deep networks on rare or ambiguous inputs, we experiment with a lightweight, simplified Confident Error Regularizer (CER) that penalizes high-confidence mistakes during fine-tuning. We summarize our contributions as follows:

- We propose and formalize the application of per-character uncertainty metrics for PD and

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 1–10
November 9, 2025

illustrate an intrinsic performance evaluation of this framework using a recent state-of-the-art neural diacritic restoration model.

- We propose an efficient calibration method and apply it on our base diacritic restoration model. We show that the approach improves uncertainty estimation at the cost of lower accuracy, while being computationally efficient.

- We discuss the potential downstream applications of such partial diacritization schemes and highlight areas that need further analysis and improvement.

2 Related Work

Early approaches to Arabic diacritization employed hidden Markov models and morphological analyzers such as MADA and MADAMIRA (Habash et al., 2005), Habash and Rambow (2005), Habash and Rambow (2007), achieving good accuracy by leveraging lexical features and large amounts of data. With the advent of neural networks, recurrent architectures and encoder–decoder transformers improved DER to below 5%. The Character-based Arabic Tashkeel Transformer (CATT) (**?**) is one such innovation, achieving very good results on both its variants, encoder only (EO) and encoder-decoder (ED).

Partial diacritization has been studied from linguistic and machine learning perspectives with the aim of improving both NLP systems as well as improving text readability. Rule-based schemes target case endings or homograph disambiguation, while supervised methods train classifiers to decide which positions to diacritize. Diab et al. (2007) investigated the impact of various diacritization schemes on Statistical Machine Translation (SMT) from Arabic to English. The authors explored different levels of partial diacritization and found that partial diacritization could improve translation quality by reducing ambiguity without significantly increasing vocabulary size or out-of-vocabulary rates. Alqahtani et al. (2016) demonstrated improvements in machine translation by employing partial diacritization strategies targeting syntactic clarity. Building on these findings, the authors also employed selective diacritic restoration specifically for homograph disambiguation (Alqahtani et al., 2019). Fadel et al. (2019) further advanced PD research by achieving state-of-the-art results and seamless integration into machine translation workflows. Qin et al. (2021) introduced regularized decoding and adversarial training to improve diacritization robustness and accuracy. Recently, Elgamal et al. (2024) analyzed naturally occurring instances of partial diacritics across diverse text genres, creating practical datasets for enhanced real-world applications.

Our contribution in relation to related work Existing PD methods depend on heuristics or linguistic context to identify words or characters for partial diacritization. None of the existing approaches leverage model uncertainty estimation techniques, which have been shown to be instrumental in other areas of application, such as computer vision (Kendall and Gal, 2017), (Lee et al., 2018), and machine translation (Pereyra et al., 2017). In this paper, we introduce the application of uncertainty estimation methods for Arabic diacritization and contribute a preliminary exploration of uncertainty metrics and performance in terms of diacritic error rates. Through this exploratory analysis, we present a case for the potential of uncertainty estimation as a viable computational approach towards partial diacritization.

3 Methodology

Our methodology for partial diacritization is to use model uncertainty to guide the removal or retention of diacritics based on target criteria. For example, if we have a fully diacritized text and wish to minimize the diacritics for improved readability, uncertainty scores may be helpful in identifying which diacritics to retain by keeping the ground truth diacritics in places with high model uncertainty. In applications where a diacritic restoration model is used directly to annotate undiacritized text, we may wish to remove predicted diacritics with high uncertainty and maintain low diacritic error rates in the resulting text. Our methodology and preliminary analysis enable both types of application by exploring the relationship between uncertainty scores and diacritic error rates. In the following sections, we describe the base model, uncertainty metrics, and the calibration scheme used to improve uncertainty estimation for diacritic restoration.

3.1 Task Formulation and Diacritic Restoration Models

Arabic diacritic restoration can be cast as a sequence-labeling problem. Given an undiacritized character sequence $\mathbf{x} = (x_1, \ldots, x_n)$, we pre-

dict a diacritic sequence $\mathbf{y} = (y_1, \ldots, y_n)$, where $y_i \in \mathcal{V}_{\text{diac}}$ (including a "no-diacritic" symbol) is a label for x_i.

We use a recent character-based transformer model for diacritic restoration, CATT (**?**), which supports both encoder-only and encoder-decoder configurations. We primarily use the encoder-decoder model in experiments, as it's shown to perform better in **?**. Both architectures are described below.

Encoder-Only (EO) is a transformer encoder θ_{enc} with a linear classification head for sequence labeling with parameters $\mathbf{W}_{\text{cls}}$ and a bias term $\mathbf{b}_{\text{cls}}$. Each position is classified independently conditioned on the entire input:

$$\mathbf{h} = \text{Encoder}(\mathbf{x}; \theta_{\text{enc}}) \quad (1)$$

$$p(y_i|\mathbf{x}) = \text{softmax}(\mathbf{W}_{\text{cls}}\mathbf{h}_i + \mathbf{b}_{\text{cls}}) \quad (2)$$

Encoder-Decoder (ED) is a full transformer architecture with autoregressive decoding. We denote the parameters of the decoder as θ_{dec}. We view the task as monotonic character-to-diacritic translation with the standard autoregressive factorization:

$$\mathbf{h}_{\text{enc}} = \text{Encoder}(\mathbf{x}; \theta_{\text{enc}}) \quad (3)$$

$$\mathbf{h}_{\text{dec}} = \text{Decoder}(\mathbf{y}_{<i}, \mathbf{h}_{\text{enc}}; \theta_{\text{dec}}) \quad (4)$$

$$p(y_i|\mathbf{x}, \mathbf{y}_{<i}) = \text{softmax}(\mathbf{W}_{\text{cls}}\mathbf{h}_{\text{dec},i} + \mathbf{b}_{\text{cls}}) \quad (5)$$

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{n} P(y_i \mid \mathbf{x},\, y_1, \ldots, y_{i-1})\,. \quad (6)$$

3.2 Uncertainty Scores

We denote the model's categorical probability output at a given character position by $p(y \mid x)$ and all parameters of the model as θ. The **Softmax Response (SR)** uncertainty is defined as:

$$U_{\text{SR}}(x) = 1 - \max_y P(y|x, \theta). \quad (7)$$

Softmax Response, (Hendrycks and Gimpel, 2017) measures model confidence explicitly, and is usually a simple baseline for alaetoric uncertainty: the inherent ambiguity of a task due to noise or multiple valid answers.

To capture epistemic uncertainty, we apply **Monte Carlo dropout** at inference time over T stochastic forward passes, obtaining distributions $p_t(y \mid x)\ t = 1 \ldots T$. This score is the difference between predictive entropy and expected entropy, termed **BALD** (Bayesian Active Learning by Disagreement) (Houlsby et al., 2011):

$$U_{\text{BALD}}(x) = H[P(y|x, \theta)] - \mathbb{E}_{q(\theta)}H[P(y|x, \theta)] \quad (8)$$

where:

- $H[P(y|x, \theta)]$ is the total uncertainty (entropy of the predictive distribution),
- $\mathbb{E}_{q(\theta)}H[P(y|x, \theta)]$ is the expected entropy over the posterior distribution of the model parameters, capturing the irreducible (aleatoric) uncertainty.

A higher BALD score indicates greater disagreement among stochastic forward passes, meaning the model lacks knowledge and would benefit from additional training on similar samples.

Finally, **Mahalanobis Distance (MD)** (Lee et al., 2018) is computed on the penultimate layer features $f(x) \in \mathbb{R}^d$, with a precomputed centroid for the whole training set μ and a covariance matrix Σ:

$$U_{\text{MD}}(x) = \sqrt{(f(x) - \mu)^T \Sigma^{-1} (f(x) - \mu)}. \quad (9)$$

Higher MD values indicate that a sample may be out-of-distribution, suggesting that the model has not encountered similar instances during training. For example, a rarely-used Arabic word or a foreign loanword transcribed in Arabic script could have a high MD score. MD is a strong epistemic uncertainty metric, since uncertainty in these instances is due to complete lack of representation rather than ambiguity.

3.3 Selective Diacritization

We propose uncertainty-based partial diacritization as follows. At inference time, we compare the uncertainty for each character position, $U(x)$, against a pre-defined threshold, τ. Depending on our objective, we can:

- Retain high confidence diacritics, where $U(x) < \tau$. This can be applied in settings where automatic diacritic restoration is used to annotate undiacritized text, and highly accurate partial diacritization is preferred over full diacritization.
- Retain low confidence diacritics, where $U(x) > \tau$. This can be used for applications where ground truth diacritics are available, and partial diacritics are sought to identify ambiguous words; for instance, in reading applications to help casual readers disambiguate difficult, ambiguous cases while maintaining minimal diacritics overall to reduce cognitive load. This approach could also be used to identify which subset of diacritics to manually annotate in an active learning framework.

By sweeping τ over $[0, 1]$, we can trace a DER–coverage curve that illustrates the trade-off between error rate and annotation effort.

3.4 Calibration via Confident Error Regularizer

Deep models often assign high confidence to incorrect predictions, which could compromise the application of uncertainty in PD as described above. To address this, we augment the standard cross-entropy loss L_{CE} with a penalty on high-confidence errors.

Xin et al. (2021) proposed the Confident Error Regularizer (CER) to add a penalty for an instance with a bigger loss than other instances and, at the same time, bigger confidence:

$$\mathcal{L}_{\text{CER}} = \sum_{i,j=1}^{k} \Delta_{i,j}\, \mathbb{I}[e_i > e_j] \tag{10}$$

$$\Delta_{i,j} = \left(\max\{0, \max p_i^c - \max p_j^c\}\right)^2 \tag{11}$$

where k is the number of instances in a batch and e_i is an error of the i-th instance: e_i is 1 if the prediction of the classifier matches the true label, and is 0 otherwise. p_i and p_j are the probabilities of these specific datapoints. The authors evaluate this type of regularization only in conjunction with the SR baseline to good results. CER is based on the principle that a well-calibrated model should assign lower confidence to incorrect predictions than to correct ones, and vice versa.

In our implementation, we adopt a simplified version of the CER that maintains the core concept while reducing computational complexity. Instead of using pairwise comparisons between all instances in a batch as in the original formulation, our approach directly penalizes high confidence on incorrect predictions with high confidence only:

$$\mathcal{L}_{\text{CER}} = \frac{\sum_{i=1}^{n} \max(p_i) \cdot \mathbb{I}[y_i \neq \hat{y}_i] \cdot m_i}{\sum_{i=1}^{n} \mathbb{I}[y_i \neq \hat{y}_i] \cdot m_i + \epsilon} \tag{12}$$

where n is the total number of tokens, $\max(p_i)$ is the maximum probability (confidence) for token i, $\mathbb{I}[y_i \neq \hat{y}_i]$ is an indicator function that equals 1 when the prediction is incorrect and 0 otherwise, m_i is a mask to ignore padding tokens, and ϵ is a small constant to avoid division by zero.

This regularization loss is then added to the task-focused cross-entropy loss $\mathcal{L}_{\text{CE}}$. The additive total loss function is then:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{CER}}. \tag{13}$$

with λ as a regularization strength hyperparameter.

4 Experiments

This section presents exploratory and experimental analysis of our approach. We analyze the performance of different uncertainty estimation methods, with a particular focus on SR as our primary score, a choice we justify below. We also evaluate the impact of confidence calibration through CER, not to be confused with Character Error Rate, with different regularization strengths. The analysis addresses several key aspects: the relationship between uncertainty thresholds and diacritization coverage, error rates at different thresholds of uncertainty, the effectiveness of uncertainty in identifying difficult words, and the calibration quality of the model with and without CER.

4.1 Datasets

The **Tashkeela Corpus (Zerrouki and Balla, 2017)** is the primary dataset usedfor training the base CATT model. We use it for fine-tuning and in-domain evaluation. The dataset contains over 75 million words of fully diacritized text, derived from classical Arabic books, religious texts, and modern Arabic educational material. We apply filtering to remove lines with less than a 60% diacritization ratio for finetuning. Since Tashkeela is a large dataset, we only fine-tuned on 10% of the data, split into an 80/20 for fine-tuning and validation.

ArVoice (Toyin et al., 2025b) is a multi-speaker Modern Standard Arabic (MSA) speech corpus with fully diacritized text transcriptions, intended for multi-speaker speech synthesis. The complete corpus consists of a total of 83.52 hours of speech across 11 voices. Since most of the ArVoice text is derived from Tashkeela, we use only the ASC subset, which is derived from the Arabic Speech Corpus (Halabi, 2016). This serves as a challenging out-of-domain test set.

4.2 Base Model without Regularization

To start the analysis, we will go over some studies on the base model itself to establish a few key points and trends, then move on to the calibration effect on key metrics, and what insights can be pulled from those differences.

4.2.1 Relationship Between Uncertainty Threshold and Diacritic Coverage

As we increase the threshold τ used to retain diacritics in the base model, we keep more diacritics, in the case that we choose to keep the ones below

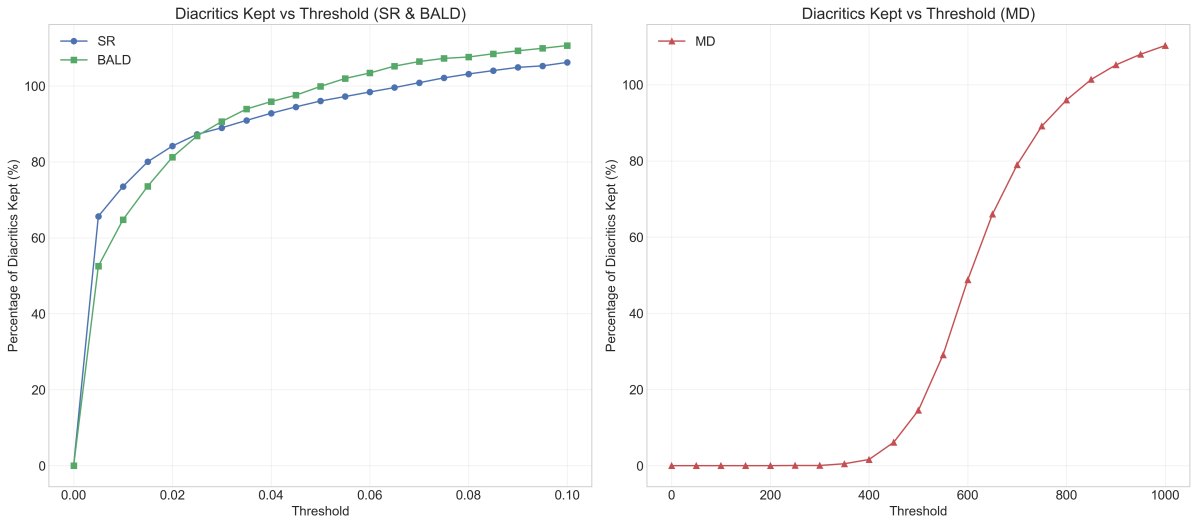

Figure 1: Percentage of diacritics kept vs. uncertainty threshold for SR and BALD (left) and MD (right) using the base model. Results are based on the **Tashkeela test set**. We plot MD separately due to the large difference in scale compared to SR and BALD. **Note**:The percentage is calculated with the total ground truth diacritics as denominator. The base model generates more than 100% of the diacritics due to insertion errors.

τ. We illustrate the relationship between uncertainty thresholds and the percentage of diacritics retained in Figure 1. Naturally, the figures follow a cumulative pattern where all diacritics are kept at the maximum uncertainty threshold.[1] The figures help identify the threshold values needed to retain a specific percentage of diacritics.

4.2.2 Relationship Between Error Rate and Diacritics Kept

While the previous section demonstrates how to select a threshold based on the percentage of desired diacritics, a more practical approach is to select a threshold based on optimal diacritic coverage and error rates. Figure 2 shows the relationship between the percentage of diacritics retained using the base model, and the resulting Diacritic Error Rate (DER) for our three uncertainty estimation metrics: SR, BALD, and MD. Fewer diacritics are favorable in a practical reading setting, since there is less visual noise to go through and less disambiguation needed. As such, keeping the smallest number of diacritics possible while retaining the lowest Diacritic Error Rate "DER" is desirable in this context. We calculate DER relative to the number of total diacritics kept.

As the illustration does not rely on absolute threshold value, we gain the advantage of visualizing the three metrics in the same scale. In the same figure, one can see that SR and BALD exhibit similar trends, with a gradual increase in error rate as more diacritics are kept. At 80% diacritization coverage, both methods maintain a relatively low error rate (approximately 2.5%, or 50% absolute reduction in error rates) after removing 20% of diacritics that have high uncertainty in the base model. In contrast, MD shows a sharper increase in error rate, suggesting that it may be unsuitable for this task. This indicates that the model's confidence (as measured by SR and BALD) is well correlated with its accuracy, making it an effective guide for partial diacritization. SR seems less prone to errors than BALD, though not significantly. SR is also much faster to compute than BALD in our encoder-decoder diacritic restoration model since MC dropout passes need to be computed for every token the decoder generates, leading to huge computational overhead. As such, SR will be chosen as the main metric of focus in the remaining analysis due to its computational efficiency and good correlation with error rates.

4.3 Confident Error Regularization

While the base model is shown to be effective at identifying many errors through uncertainty scores, effectively reducing error rates by 50% while maintaining 80% of diacritics, we still have many instances where the model uncertainty scores do not track performance, especially in the out-of-

[1]The model predicts more diacritics than the reference ground truth, making the results go above 100% at the extremes due to insertions it makes.

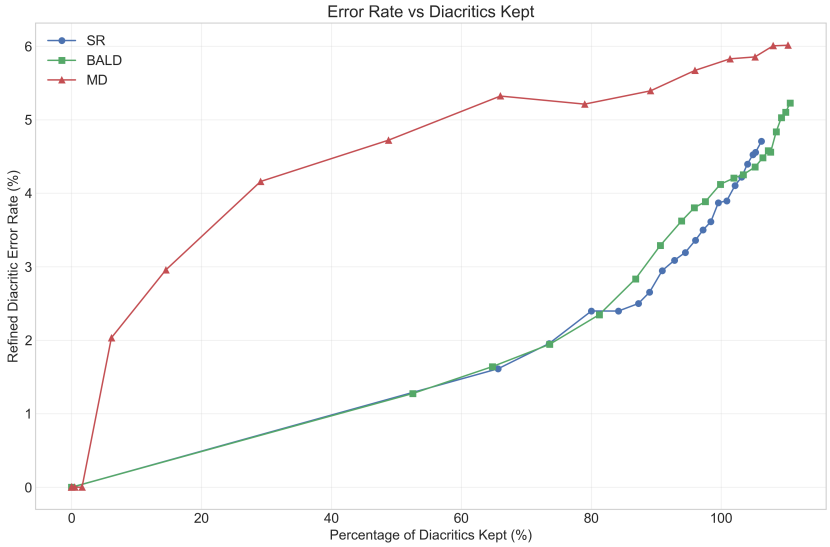

Figure 2: Diacritic Error Rate vs. Percentage of Diacritics Kept on the **Tashkeela** test set for SR, BALD, and MD metrics using the base model, keeping diacritics below threshold. **Note**: The percentage is calculated with the total ground truth diacritics as denominator. The base model generates more than 100% of the diacritics due to insertion errors.

domain test set. To address this calibration issue, we applied CER with different regularization strengths λ. We experimented with several values, selected through hyperparameter optimization for confidence gaps and validation diacritic error rate. The results below are shown using the out-of-domain test set derived from ArVoice, where the base error rate is above 10%.

4.3.1 Impact on DER

Diacritic error rate tends to increase with higher regularization, as shown in Figure 3.

This increase is more significant for $\lambda_{2.77}$, and scales mostly linearly as λ values increase. $\lambda_{0.644}$ shows moderate increase in DER, while improving model precision in detecting error-prone, high-DER words, as discussed in the next section.

4.3.2 High-DER Words Precision/Recall

To quantify the model's ability to identify ambiguous words, we perform word-level analysis. We define a high-DER word as one with $> 50\%$ DER. We then measure how well we can identify these high-DER words using the model's uncertainty scores. We calculate the uncertainty score for a word as the mean uncertainty of its characters. We then sort the words in the test set from lowest to highest uncertainty to define the uncertainty percentiles. For instance, the 70^{th} percentile is the word-level uncertainty score where 70% of the words fall below, and 30% of words are higher. The 30% high-uncertainty words are the ones 'detected' by the model. The exact calculations are shown in Appendix, section A.1.2.

Based on these definitions, we measure the precision and recall at different regularization strengths λ and uncertainty percentiles[2]. Figure 4 presents these metrics for several values of the regularization parameter λ, along with the base model ($\lambda = 0$). Overall, there is a clear trade-off between recall and precision. The base model tends to achieve higher recall but suffers from very low precision, indicating that it flags words with low error rates as uncertain, and vice versa. In contrast, the regularized models typically flag fewer words overall, which results in smaller recall but precision is higher than the baseline.

Notably, at very high thresholds (e.g., the 99th or 100th percentile), both recall and precision drop, likely because only a tiny fraction of words exceed these stringent uncertainty levels. A threshold near the 90-95% range appears to offer a good balance between detecting enough erroneous words while minimizing false positives. The exact choice depends on whether higher precision or higher coverage of erroneous words is the primary goal.

[2]Note that due to the distribution of the scores and the skewed uncertainty values, the percentiles do not reflect exactly the same number of detected words across models. For instance, at the 70^{th} percentile, the number of words below the threshold may be less than 70%.

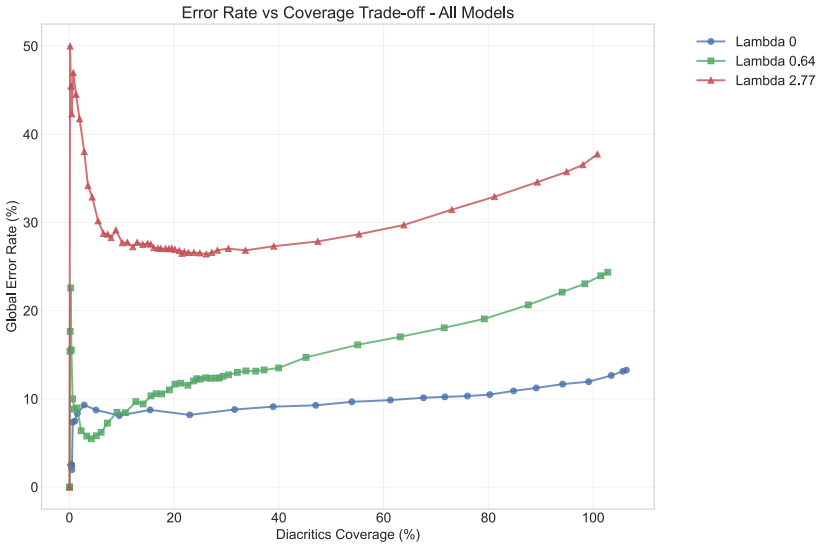

Figure 3: DER calculations for the base and calibrated models across coverage levels. Keeping below threshold diacritics. The results are based on the **ArVoice test set**. **Note**: The coverage is calculated with the total ground truth diacritics as denominator. The base model generates more than 100% of the diacritics due to insertion errors.

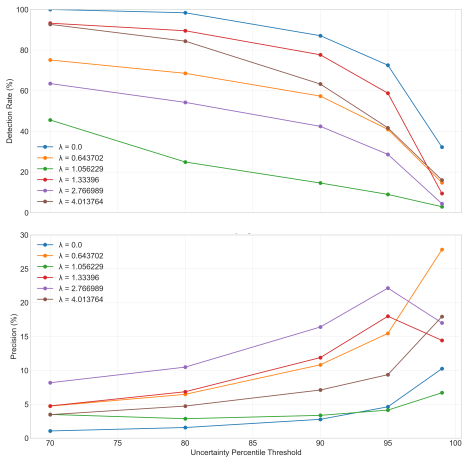

Figure 4: Detection rate or recall (top) and precision (bottom) of high-DER words at different uncertainty percentiles using different regularizing strength values. The results are based on the **ArVoice test set**.

5 Discussion and Conclusions

We explored the application of uncertainty estimation for partial diacritization of Arabic text. We experimented with various uncertainty estimation methods, and established the suitability of Softmax Response for this task. The other two metrics we explored had some drawbacks that made them less suitable for our task. While the Mahalanobis distance exhibited some correlation with diacritic error rates, the effect is weaker than the other two methods, resulting in higher error rates at the same coverage points. BALD achieved similar correlation to SR, but it is less suitable for practical diacritc restoration models that involve sequence labeling due to its higher computational cost. Among the three metrics, SR provides optimal performance and efficiency, making it suitable for additional calibration and practical deployment. However, our experiments show that better calibration results in higher DER, so additional work is needed to develop calibrated models that retain base accuracy. Nevertheless, the calibrated model shows potential for identifying ambiguous words, which we define as words with high DER, in terms of precision. This indicates that calibration may still be useful for some target application, where identifying ambiguous words with high precision (albeit with low recall) is desired. This preliminary analysis illustrates that additional work is needed to identify suitable calibration methods that optimize uncertainty estimation while maintaining the performance of the base diacritic restoration model.

In terms of application, the SR-based approach is straightforward to integrate into any neural-based diacritic restoration model, and our experiments show that we can reduce the relative DER in partial diacritization with various coverage thresholds. Such approach can be used in user-facing applications where automatic diacritization is used to annotate undiacritized text, leading to a partially diacritized text that is more accurate than the baseline. However, such methods do not address the issue of ambiguous words, which are likely to re-

Table 1: Examples of Threshold-based Diacritic Selection at 20% Coverage ($\lambda = 0.64$)

Strategy	Text
Ground Truth	الْماءُ يَتَجَمَد عِنْدَ أَكْثَرَ مِنْ مِئَةِ دَرَجَةٍ فِي ظِلِ تَنَامِي الْآثَارِ الْجَانِبِيَة لِلْأَدْوِيَة الْكِيمْيَائِيَّة
Drop High Uncertainty	الماء يتجمد عنْدَ أَكثَر مِن مئَةِ دَرَجَة في ظِل تنَامِي الآثَارِ الْجانِبِيَة لِلأدوِيةِ الكِيمِيَائِيَة
Drop Low Uncertainty	الْماءُ يتَجمَد عِند أَكْثَرَ منْ مِئَة درجةٍ فِي ظلّ تَنامي الْآثار الجَانبية للْأَدْوِيَة الْكيميائيَة

main undiacritized under such schemes. For improving readability, the same technique could be used to reduce the total number of diacritics in fully-diacritized text with ground-truth diacritics. The words that are identified as high-uncertainty could retain their diacritics, while diacritics on low-uncertainty words can be dropped. Examples of sentences and their partial diacritics using each of these proposed schemes are shown in Table 1.

Our exploratory analysis provides a starting point for such applications; further evaluation and analysis are needed to verify the effectiveness of such approaches in practical applications like readability enhancement, machine translation, and text-to-speech synthesis.

Limitations

We limited our analysis to one base diacritic restoration model, CATT, which serves as a strong baseline. Our analysis may be applicable to other models, but the experiments need to be replicated to verify that. The work presented in this paper serves as a preliminary exploration of uncertainty estimation as applied to the task of diacritic restoration, but it does not include sufficient analysis of the impact of such methods on downstream applications. Additional experiments are needed to explore the applicability of the proposed technique in applications such as machine translation, text-to-speech synthesis, or readability assessment. The choice of uncertainty metrics was motivated mostly by simplicity and convenience, and other metrics could have been included in the analysis. The analysis provided in this paper should be taken as a partial exploration rather than the final word on the suitability of uncertainty estimation metrics for partial diacritization. Finally, the experiments show that error calibration hurts model performance. We do not provide a solution for this and leave any improvement on the proposed calibration method for future work.

Acknowledgements

We would like to thank members of the speech lab at MBZUAI, namely Hawau Olamide Toyin, and Rufael Fekadu Marew, for guidance and support on various steps in this work. We also thank the anonymous reviewers at UncertaiNLP, who engaged constructively with the paper and raised valid points. We added these points to the limitations section.

References

Sawsan Alqahtani, Hanan Aldarmaki, and Mona Diab. 2019. Homograph disambiguation through selective diacritic restoration. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 49–59, Florence, Italy. Association for Computational Linguistics.

Sawsan Alqahtani, Mahmoud Ghoneim, and Mona Diab. 2016. Investigating the impact of various partial diacritization schemes on Arabic-English statistical machine translation. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track*, pages 191–204, Austin, TX, USA. The Association for Machine Translation in the Americas.

Ali Assad, Abdul Hadi M. Alaidi, Amjad Yousif Sahib, Haider TH Salim ALRikabi, and Ahmed Magdy. 2024. Transformer-based automatic arabic text diacritization. *Sustainable Engineering and Innovation*, 6(2):285–296.

Mona Diab, Mahmoud Ghoneim, and Nizar Habash. 2007. Arabic diacritization in the context of statistical machine translation. In *Proceedings of Machine Translation Summit XI: Papers*, Copenhagen, Denmark.

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.

Muhammad ElNokrashy and Badr AlKhamissi. 2024. A context-contrastive inference approach to partial diacritization. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 89–101, Bangkok, Thailand. Association for Computational Linguistics.

Ali Fadel, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019. Neural Arabic text diacritization: State of the art results and a novel approach for machine translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 215–225, Hong Kong, China. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2007. Arabic diacritization through full morphological tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56, Rochester, New York. Association for Computational Linguistics.

Nizar Habash, Owen Rambow, and George Kiraz. 2005. Morphological analysis and generation for Arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 17–24, Ann Arbor, Michigan. Association for Computational Linguistics.

Nawar Halabi. 2016. Arabic speech corpus.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jurgen Lameris. 2021. Homograph disambiguation for arabic text-to-speech synthesis using transformer-based models. Master's thesis, Uppsala University.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7167–7177.

Hamdy Mubarak, Ahmed Abdelali, Hassan Sajjad, Younes Samih, and Kareem Darwish. 2019. Highly effective Arabic diacritization using sequence to sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2390–2395, Minneapolis, Minnesota. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, ukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *International Conference on Learning Representations (ICLR)*.

Han Qin, Guimin Chen, Yuanhe Tian, and Yan Song. 2021. Improving Arabic diacritization with regularized decoding and adversarial training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 534–542, Online. Association for Computational Linguistics.

Mark Roman and Bernard Pavard. 1987. Processing diacritics in reading arabic: Evidence from eye-movement measures. In *Proc. of the Annual Meeting of the Cognitive Science Society*. Cognitive Science Society.

Hawau Olamide Toyin, Samar M Magdy, and Hanan Aldarmaki. 2025a. Are LLMs good text diacritizers? an Arabic and Yorùbá case study. *arXiv preprint arXiv:2506.11602*.

Hawau Olamide Toyin, Rufael Marew, Humaid Alblooshi, Samar M Magdy, and Hanan Aldarmaki. 2025b. ArVoice: A Multi-Speaker Dataset for Arabic Speech Synthesis. In *Interspeech 2025*, pages 4808–4812.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051.

Taha Zerrouki and Amar Balla. 2017. Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems. *Data in Brief*, 11:147–151.

A Appendix

A.1 Formulation of metrics

We use the following metrics to evaluate our models, categorized into diacritic-level and word-level metrics:

A.1.1 Diacritic-level Metrics

1. **Diacritic Error Rate (DER)**: The percentage of incorrectly predicted diacritics over the total:

$$\text{DER} = \frac{|\{d \in \mathcal{D} | \hat{d} \neq d\}|}{|\mathcal{D}|} \times 100\% \quad (14)$$

Where:

- d is a given diacritic
- $\mathcal{D}$ is the set of all possible diacritics
- $\hat{d}$ is a true label diacritic

This metric is computed using edit distance calculations adapted for Arabic diacritics. Any missing or deleted diacritics are not considered errors, since the goal is to remove as many diacritics as possible while retaining accurate predictions by the model.

2. **Diacritization Coverage**: Percentage of characters or words that retain diacritics after partial diacritization:

$$\text{Coverage} = \frac{|\{c \in \mathcal{C} | \text{HasDiac}(c) = \text{True}\}|}{|\mathcal{C}|} \times 100\% \quad (15)$$

Where:

- c is a given character
- $\mathcal{C}$ is the set of all characters that can be diacritized
- 'HasDiac' is a function that returns True when character c retains its diacritic after thresholding, and False otherwise

Controlled by uncertainty threshold, Lower coverage means more sparsely populated text (fewer diacritics).

A.1.2 Word-level Metrics

1. **High/Low DER Words**: A heuristic definition of high-and low-DER words. Words with Diacritic Error Rate (DER) exceeding 50% are defined here to be able to see the model's consistency in capturing such error-prone words. We define the set of high DER words as:

$$\mathcal{H} = \{w \in \mathcal{W} | \text{DER}(w) > 0.5\} \quad (16)$$

And the set of low DER words as:

$$\mathcal{L} = \mathcal{W} \setminus \mathcal{H} = \{w \in \mathcal{W} | \text{DER}(w) \leq 0.5\} \quad (17)$$

The uncertainty of a word is calculated as the mean uncertainty across all characters in the word:

$$U(w) = \frac{1}{|w|} \sum_{j=1}^{|w|} U(c_j) \quad (18)$$

Where:

- w represents any given word
- c_j represents the j-th character in word w
- U is the uncertainty of word w or character c_j
- $\mathcal{W}$ is the set of words in total in the dataset

2. **Recall and precision**:

Recall is the percentage of detected high-DER words over their total amount, detected or not:

$$\text{Recall} = \frac{|\{w \in \mathcal{H} | \text{Detected}(w) = \text{True}\}|}{|\mathcal{H}|} \quad (19)$$

Precision is the amount of the detected high-DER words, over the total detected words by the model:

$$\text{Precision} = \frac{|\{w \in \mathcal{H} | \text{Detected}(w) = \text{True}\}|}{|\{w \in \mathcal{W} | \text{Detected}(w) = \text{True}\}|} \quad (20)$$

Effectively, recall shows us how good a model is at catching problems in general, how much it can actually cover of them in total, and precision shows us how accurately it can catch actual, legitimate ambiguous cases, rather than flagging any given word overall as uncertain with inflated scores