Uncertainty in Semantic Language Modeling with PIXELS

Stefania Radu, Marco Zullich, Matias Valdenegro-Toro
Department of AI, Bernoulli Institute, University of Groningen, The Netherlands
stefania.m.radu@gmail.com, m.a.valdenegro.toro@rug.nl

Abstract

Pixel-based language models aim to solve the vocabulary bottleneck problem in language modeling, but the challenge of uncertainty quantification remains open. The novelty of this work consists of analysing uncertainty and confidence in pixel-based language models across 18 languages and 7 scripts, all part of 3 semantically challenging tasks. This is achieved through several methods such as Monte Carlo Dropout, Transformer Attention, and Ensemble Learning. The results suggest that pixel-based models underestimate uncertainty when reconstructing patches. The uncertainty is also influenced by the script, with Latin languages displaying lower uncertainty. The findings on ensemble learning show better performance when applying hyperparameter tuning during the named entity recognition and question-answering tasks across 16 languages.

1 Introduction

After the release of ChatGPT in 2022, the number of papers published every day on the topic of Large Language Models (LLMs) has increased more than 20-fold (Zhao et al., 2023). The number of parameters in these models jumped from 340 millions in implementations such as BERT (Devlin et al., 2018) to billions of parameters in models like GPT-3 (Brown et al., 2020) or LLaMA (Touvron et al., 2023). Despite their obvious popularity, one of the central limitations of LLMs remains their uncertainty and lack of trustworthiness (Huang et al., 2024). As these models are being applied more and more to high-stakes scenarios, such as medicine (Busch et al., 2025) or security (Gawlikowski et al., 2023), it is critical that their predictions can be trusted. Generally, the research on the explainability and interpretability of LLMs is focused on traditional tokenizer-based methods, that split text into smaller units. They produce overconfident responses even when the predictions are likely incorrect (Xiong et al., 2023).

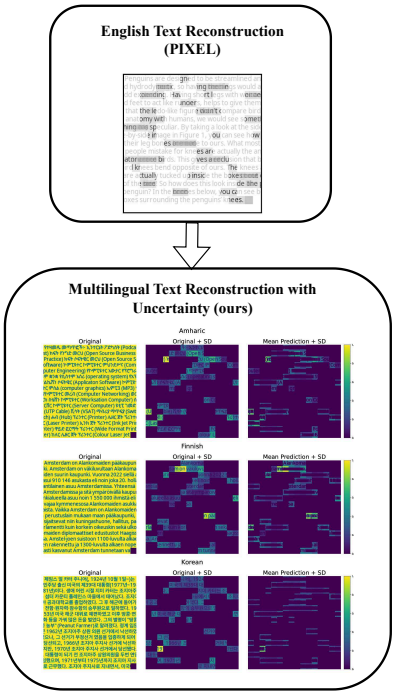


Figure 1.1: Example of text reconstruction using the PIXEL model from Rust et al. (2022), and text reconstruction with uncertainty for different languages.

For semantic NLP tasks such as extractive question answering (QA), it is common to use models that predict the start and end tokens of an answer span and provide confidence scores based on the softmax probabilities of these predictions (Devlin et al., 2018; Lan et al., 2019). However, this approach offers no measure to quantify the uncertainty of the prediction. Several works have been proposed in the past years to solve this problem (Xiao et al., 2022; Lin et al., 2023). Common solutions include incorporating uncertainty directly into the model using Bayesian Neural Networks (BNN) (Blundell et al., 2015) or post-hoc methods such as

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 103–119
November 9, 2025

Monte Carlo Dropout (Gal and Ghahramani, 2016), Temperature Scaling (Guo et al., 2017) and Ensemble Learning (Lakshminarayanan et al., 2017). However, these approaches have not been studied in the context of more recent pixel-based models that use visual representations of words, as opposed to text representations.

The *Pixel based Encoder of Language* or PIXEL proposed by (Rust et al., 2022) aims to transform language modeling into a visual recognition task with the help of small and square clusters of pixels, called *patches*. PIXEL does not rely on a predefined vocabulary and it is trained to reconstruct missing patches of text, by following a Vision Transformer – Masked Autoencoder (ViT-MAE) architecture. The Vision Transformer (ViT) uses linear embeddings of fixed-sized patches of pixels which are encoded using a transformer. In the context of computer vision, masked image encoding works similarly to masked language modeling (MLM), by masking regions of an image and then learning to reconstruct the whole image.

PIXEL was pretrained on rendered versions of the Wikipedia and BookCorpus datasets and it is evaluated on 32 topologically diverse languages, across 14 scripts. Supporting multiple languages requires a larger vocabulary to cover diverse linguistic features and scripts, which is often impractical within the constraints of a fixed vocabulary size. Wu and Dredze (2019) noted that multilingual models struggle with resource allocation across languages, leading to suboptimal performance in less represented languages, during tasks like named entity recognition, part-of-speech tagging, and dependency parsing. Furthermore, imbalanced vocabulary representation can exacerbate biases, resulting in unfair treatment of certain languages (Wan, 2021). The trade-off in vocabulary allocation means that models either inadequately repre sent some languages or become too large in size and computational requirements.

The main aim is to study uncertainty in pixel-based language models focusing on semantic tasks. Given the challenging nature of semantic processing and the fewer studies dedicated to it, this research will center on finetuning models to solve tasks like named entity recognition, sequence classification, and question answering. Solving the vocabulary bottleneck of traditional language models which rely on a close vocabulary can be achieved by using pixel-based models which do not require a fixed vocabulary. Finally, to tackle the uncertainty problem, this work will make use of existing techniques for quantifying uncertainty, and apply them to pixel-based models, which also represent the biggest novelty of this study. This includes uncertainty quantification at the pixel level using Monte Carlo methods (Figure 1.1), ensemble learning applied to models finetuned on three semantic tasks across 19 languages, but also an analysis of the attention mechanism.

2 State of the Art

The first study to use visual features of text in order to create embeddings was applied to Chinese and used linearizing bitmaps of characters or words (Aldón Mínguez et al., 2016). By using shared character components from Chinese or Korean, it becomes easier to generalize to new and less frequent characters. Different studies (Dai and Cai, 2017; Sun et al., 2018; Salesky et al., 2021) used rendering techniques to obtain images of text. In this context, text rendering involves converting character codes into glyph indices, which are then used to generate the corresponding glyph images, while applying various styles, fonts, sizes, and colors. A glyph often contains one character only, but it can also represent accents or multiple characters in languages where ligatures are common, like Arabic. Dai and Cai (2017) used text rendering in Chinese, Japanese, and Korean, and extracted visual features from a Convolutional Neural Network (CNN) to perform text classification. Similarly, Sun et al. (2018) applied convolutions to squared rendered images to perform sentiment analysis in Chinese and English.

In the context of machine translation, Salesky et al. (2021) suggested a very robust approach based on a variation of the ViT. The training data is rendered into gray-scale images using the Pygame backend and a slicing window is applied to create patches, which act as tokens. Then, a 2D convolutional block followed by linear projection is used to create embeddings, which serve as input for the transformer encoder. The translation happens directly from pixel representations, without any word preprocessing. After training on seven language pairs, the approach matches the performance of traditional language models, with additional advantages. It is more robust to character permutations or substitutions, and it does not rely on text preprocessing steps, such as tokenization or segmentation.

As of to date, systematic investigations into the

uncertainty and calibration of pixel-based language models remain limited. Rust et al. (2022) showed that PIXEL is robust when it comes to character-level perturbations and code-switching. In this analysis, relevancy heatmaps were used to depict visual explanations of correct predictions, and there is evidence to suggest that these outputs are interpretable when identifying contradictions and entailment relationships. However, during semantic tasks like named entity recognition, sequence classification, and question answering, PIXEL is struggling to retain semantic knowledge and transfer it across scripts. Reasons for this might include a lack of multilingual pretraining, as well as a limited ability to capture contextual information due to the use of unigram patch embeddings. While raw performance is desirable, it is crucial to have models that are reliable and explainable.

3 Methods

3.1 Data

MasakhaNER 1.0 MasakhaNER 1.0 (Adelani et al., 2021) is a Named Entity Recognition (NER) benchmark, which includes data from 10 African Languages obtained from local news sources (Amharic, Hausa, Igbo, Kinyarwanda, Luganda, Luo, Nigerian-Pidgin, Swahili, Wolof and Yorùbá), as well as the ConLL-2003 English dataset. The task involves classifying named entities into nine pre-defined categories. The MasakhaNER dataset contains labeled entities for each language.

GLUE The Sequence Classification (SC) task relies on the The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018). It involves nine sentence-level understanding tasks (CoLA, SST-2, MRPC, QQP, STS-B MNLI-M/MM, QNLI, RTE, WNLI) in English, across three categories: single-sentence tasks, similarity and paraphrase tasks, and inference tasks.

TyDiQA-GoldP To assess the ability of the model to perform Question Answering (QA), the TyDiQA-GoldP dataset was selected (Clark et al., 2020). It contains nine typologically diverse languages (English, Arabic, Bengali, Finnish, Indonesian, Korean, Russian, Swahili, Telugu). The dataset contains questions written by native speakers, passages with relevant information, and answers provided as short spans of text within the passage. Unlike the primary task, the Gold Passage task focuses more on locating the exact answer within a given context.

3.2 Model Architecture

PIXEL processes text as images that are rendered using the PyGame[1] renderer to accommodate multiple scripts. Each rendered image is converted into a sequence of patches, resulting in 529 non-overlapping patches, with a size of $16*16$ pixels. A ViT-based encoder encodes visible patches and the CLS tokens through patch, positional, and CLS embeddings. During pretraining, the system applies random masking to 25% of the patches and employs a decoder to reconstruct the masked regions through a regression-like method. The decoder is then finetuned on downstream tasks by replacing the reconstruction objective with task-specific heads.

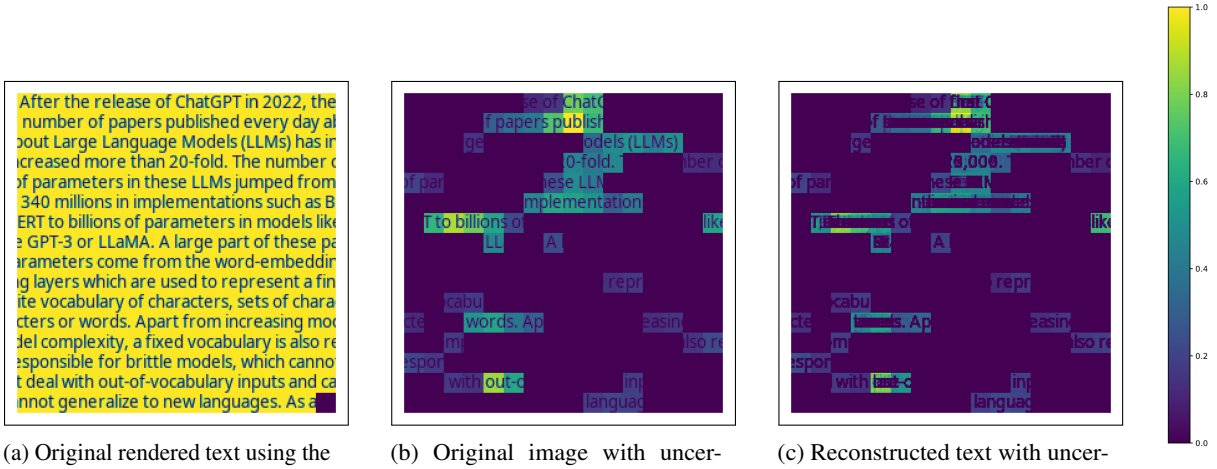
The English PIXEL which serves as a base for the experiments described in the next section is pretrained on a rendered version of English Wikipedia and BookCorpus (Zhu et al., 2015). For more details about the PIXEL pretraining routine, refer to the implementation[2] of Rust et al. (2022).

3.3 Uncertainty Quantification

Monte Carlo Uncertainty The first method used to quantify epistemic uncertainty at the patch level is Monte Carlo (MC) Dropout. The input is a rendered `image` $\in \mathbb{R}^{16\times16\times3}$ with a sequence length of 256 pixels, and the goal is to obtain an uncertainty map $U \in \mathbb{R}^{16\times16\times3}$, containing the uncertainty for each patch. For this, the model is used in 100 forward passes to compute a series of predictions P, which contain per-pixel logits. Then, the mean prediction is created by averaging these logits, resulting in the reconstructed text. A standard deviation (SD) image is obtained by computing the SDs of the predictions for each pixel. Since each patch has a dimension of 16×16 pixels, the per-patch uncertainty is defined by averaging the predictions of all SD values inside a patch, and each pixel inside the patch is assigned that value. Finally, the uncertainty map U is a collection of patches representing the overall uncertainty of its pixels. For visualization purposes, the uncertainty map is overlaid on top of the original image, as well as on the reconstructed text. An overview of this routine is presented in Algorithm 1 of Appendix C.

An overall mean uncertainty value ($\bar{\sigma}$) is also computed to measure uncertainty at the image level (Equation 3.1), where H and W refer to the height

[1] https://www.pygame.org/
[2] https://github.com/xplip/pixel



(a) Original rendered text using the PyGame renderer.

(b) Original image with uncertainty.

(c) Reconstructed text with uncertainty.

Figure 2.1: Example of uncertainty quantification at the patch level for an image containing text from the introduction of this paper. Brighter colors indicate more uncertainty.

and width of the image.

$$\bar{\sigma} = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} \sigma(h, w) \tag{3.1}$$

Additionally, we compute two loss functions during the MC inference: the normalized MSE loss (Equation 3.2) used during pretraining and the normalized Gaussian Negative Log-Likelihood (GNLL) loss (Equation 3.3), where $eps = 1e - 6$ is a clamp value used for stability. Unlike the MSE, the GNLL loss accounts for epistemic uncertainty, by incorporating the variance of the predicted distribution.

$$\text{MSE} = \frac{1}{H \times W} (\text{pred} - \text{img})^2 \tag{3.2}$$

$$\text{GNLL} = \log(\max(\text{var}, \text{eps})) + \frac{(\text{pred} - \text{img})^2}{\max(\text{var}, \text{eps})} \tag{3.3}$$

We study uncertainty across tasks: NER (MasakhaNER 1.0), SC (GLUE), and QA (TyDiQA-GoldP), and scripts – as one of the main challenges in NLP is building reliable models that can scale up to real-world applications where many scripts are often encountered. Additionally, we carry out a calibration analysis to examine the relationship between model performance and uncertainty across tasks. The performance is measured using Root Mean Square Error (RMSE $= \sqrt{\text{MSE}}$, Equation 3.2), while uncertainty is quantified using MC standard deviation. The goal is to evaluate how well the predicted uncertainty values align with actual performance errors across the different scripts and languages.

Attention Visualization To visualize attention in the PIXEL encoder, a square attention grid $A \in \mathbb{R}^{L \times H \times N_{\text{patches}}{}^2}$ is created for the encoded patches, where L is the number of attention layers and H is the number of heads in each layer. An example is presented in Figure 3.1. This shows model-level attention across all layers and heads for a particular input image. Each cell $A(l, h)$ in this grid visualizes the neuron-level attention weights for a specific head h and layer l. Then, each patch in the attention cell attends to the other patches in the sequence according to the dot product between the query (of the *attender* patch) and the key (of the *attended* patch). The weights are averaged over 100 Monte Carlo forward passes. Considering the increased dimensionality of the attention cell, only the first 16 patches are visualized, resulting in an image with 16×16 patches.

Ensemble Learning To solve the *Extractive Question-Answering* task, four learner models are finetuned on each of the 9 languages of the TyDiQA-GoldP (Section 3.1) dataset, resulting in 36 total models. Each model is trained on the `train` split of a language in the dataset and evaluated on the `validation` split of the same language. There are four main steps to be followed to compute the final prediction for an input question. In a regular non-ensemble setting, there is only one finetuned model that dictates the output answer for each example. In the ensemble learning framework, each model M_i is applied to the input question q to

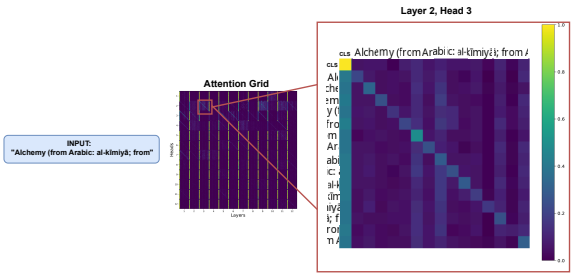


Figure 3.1: Model-level (attention grid) and neuron-level (layer 2, head 3) views of attention in the PIXEL model for a short input text from the English Wikipedia. The attention grid contains 12 attention layers with 12 attention heads each.

obtain the candidate answers with corresponding confidence probability values. To reduce the pool of candidates, only the predictions that appear in all models are kept. The average confidence conf_c is computed for each candidate across all models. Finally, the candidate with the highest confidence is selected.

In the *Named Entity Recognition task*, five learner models are finetuned on each of the 10 languages of the MasakhaNER 1.0 dataset (Adelani et al., 2021), resulting in 50 total models. Each model is trained on the `train` split of a language in the dataset and evaluated on the `test` split of the same language. The task involves assigning a label to each token from a list of 9 predefined classes. Their predicted logits are averaged and combined into one value for each class. The final label is computed as shown in Equation 3.4, where L is the set of labels (classes) and k is the number of models.

$$\text{label} = \arg\max_{l \in L} \left(\frac{1}{k} \sum_{i=1}^{k} \text{logits}_{i,l} \right) \tag{3.4}$$

During the ensemble experiment, only the values of the batch size (BSZ), learning rate (LR), dropout probability (DP), and the seed are changed. For more details about the finetuning configuration and routine, refer to Tables C.3 and C.2.

4 Results

4.1 Monte Carlo Uncertainty

Uncertainty Across Datasets The distribution of MC uncertainty is presented in Figure 4.1 (left), suggesting that GLUE achieves the highest overall uncertainty, which indicates that pixel-level uncertainty increases with text that has more semantic complexity, as it is the case in sentiment classification, semantic similarity or textual entailment tasks.

In terms of the mask ratio R, the plot indicates that lower values (0.1 to 0.3) generally correspond to lower uncertainty across all datasets, hinting that less masking leads to more certain predictions. In this case, the largest part of the data is concentrated between uncertainty values of 0.15 and 0.25. As the mask ratio increases, the distribution becomes more spread out.

The results from Figure 4.2 (left) indicate that the loss increases with the mask ratio. This is expected as the model was trained to reconstruct the image patches with a mask ratio of $R = 0.25$. There is also a wide performance gap between the sequence classification task (GLUE) and the rest of the tasks, which can be attributed to language. The GLUE dataset contains English text, the language the PIXEL model was pretrained on, while TyDiQA-GoldP and MasakhaNER are multilingual datasets.

Uncertainty Across Scripts The overall trends (right) show that Ge'ez, Chinese Characters, Arabic, and Korean scripts exhibit high uncertainty (Figure 4.1, right) and high mean loss (Figure 4.2, right), and the increase is more pronounced at mask ratios above 0.6. The Latin and Cyrillic scripts are increasing more gradually with a sharper uptick around 0.8 – 0.9. The main script found in the pre-training datasets (English Wikipedia and the BookCorpus) is Latin, and there is a high overlap between Latin and Cyrillic characters, given that both scripts share Greek as a common ancestor. However, the uncertainty in the Cyrillic script is lower, compared to Latin. The scripts with the highest MC uncertainty are Ge'ez and Chinese Characters, both of which are visually quite distinct from the Latin script.

Calibration Analysis To further study the relationship between performance and uncertainty, Figure 4.3 depicts a hexbin plot with marginal distributions, where the Root Mean Squared Error (RMSE) loss is plotted against the SD uncertainty from the MC experiments. The x-axis represents the aggregated per-image standard deviation (uncertainty) of the model after 100 Monte Carlo samples. The RMSE measures the average of the actual errors between the true pixel values and the predicted values. Inside each hexagon, the color intensity corresponds to the density of data points within that hexagon. Therefore, darker regions indicate a

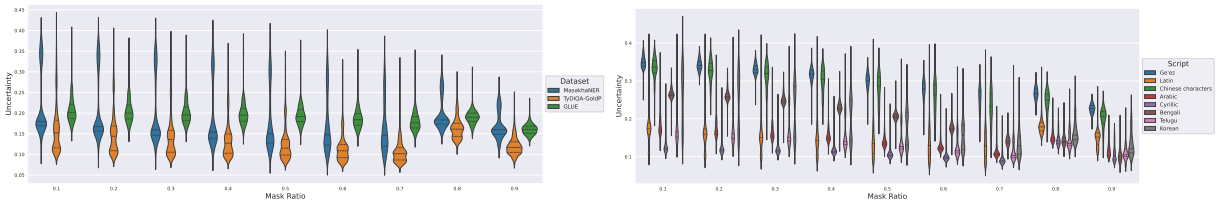


Figure 4.1: The distribution of the MC Uncertainty across the different datasets (left) and scripts (right) for each mask ratio value R.

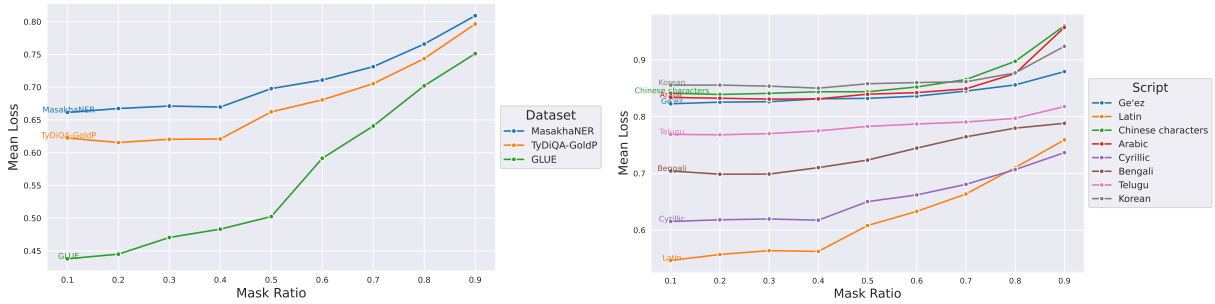


Figure 4.2: The MSE loss across the different datasets (left) and scripts (right) for each mask ratio value R.

higher density of data points. There is a high density of points in the top left corner, which suggests that the model underestimates its performance. In other words, many examples are associated with high loss but low uncertainty.

The distribution of the points for all three datasets (MasakhaNER, TyDiQA-GoldP, and GLUE) is shown in the calibration plot from Figure 4.4. The highest level of overconfidence is associated with the question-answering task in TyDiQA-GoldP. However, there seems to be a subgroup of points for which the uncertainty is high. The points in the MaskhaNER dataset fall under the category of high uncertainty and high loss. The GLUE data is located between 0.15 and 0.3 on the uncertainty range and contains several examples showing decreased loss. While the model can be considered to be underestimating uncertainty with this group, the majority of the data still fall over the main diagonal, indicating an underestimation of uncertainty.

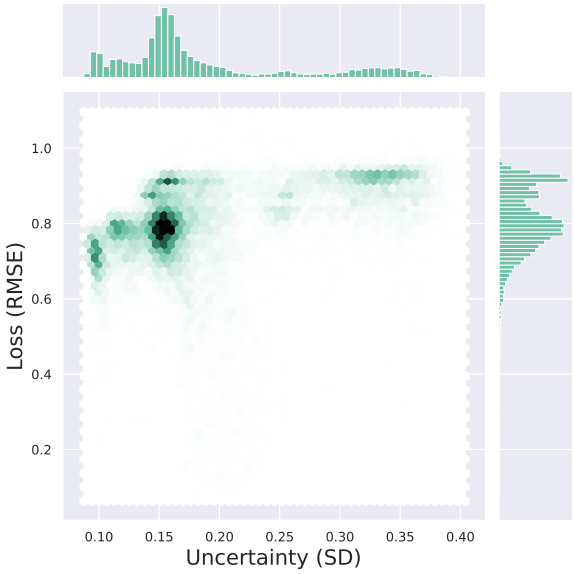


Figure 4.3: Calibration hexbin plot showing the RMSE loss in terms of the MC uncertainty.

Visualizing Uncertainty in Text Reconstruction Figure 2.1 shows (a) the original rendered English text generated with the PyGame text renderer, (b) the original image overlaied with per-patch uncertainty and (c) the reconstructed text overlaied with per-patch uncertainty. Bright yellow patches suggest larger variations in predictions. This can be observed in the larger masked segments of patches from the first 6 lines of the image, as well as in lines 12 and 15. These segments also translate to less accurate reconstructions, as seen on the corresponding rows of the reconstructed image. On the other hand, smaller segments of patches (which appear darker in the image) are associated with lower uncertainty and are reconstructed more accurately. These patches often contain shorter sequences of letters. In terms of the mistakes, the model fails to reconstruct patches with numerals, such as *20-fold*. Still, it appears to understand that the most suitable prediction given the context is a number (the model predicts *20,000*). Moreover, longer and less frequent words such as *implementation* and *pub-*

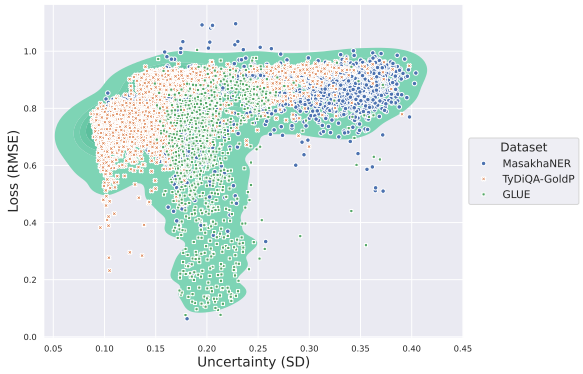


Figure 4.4: Calibration kernel density estimate plot showing the RMSE loss in terms of the MC uncertainty across the three datasets.

lish, as well as punctuation marks (used in *(LLMs)*) appear to produce more variation in the prediction, given the increased uncertainty.

4.2 Attention Visualization

Each cell in the attention grids (Figure 4.5) shows the attention weights for the first 16 patches of a specific head h and layer l in the selected examples. The first four layers appear to encode the highest amount of visual information, given the high activation of the patches. Across all heads and layers of both examples, the attention weight corresponding to the CLS patch is high, as it contains the aggregate representation of the input patch sequence. There is a clear difference in the distribution of attention between the examples. The top 1 performer (Nigerian Pidgin) exhibits high activation on the diagonal at the neuron level, meaning that patches are attending to themselves, possibly to retain positional and contextual information. The Igbo example does not show the same pattern, rather a subset of dominant patches attend to the remaining ones.

4.3 Ensemble Learning

Extractive Question Answering The results of the ensemble QA model are presented in Table 4.1, which shows the weighted F1 score across all languages in the TyDiQA-GoldP dataset. These findings are compared with the results obtain by Rust et al. (2022), following the same experimental setting. Overall, the ensemble learning method improves the performance in the extractive QA task for 6 out of the 8 languages. The average F1 score (excluding the ENG data) for the ensemble configuration is higher with 1.7 points than in the case of the regular PIXEL model. In terms of the individual languages, there is a high improvement for Indonesian (4.3 points), Russian (2.8 points), and Arabic (2.2 points), suggesting that combining multiple learners can improve performance regardless of script.

Figure 4.6 presents the confidence distribution of the best answers in the ensemble model for all languages in the dataset. In general, the confidence is in the range $0.2 - 0.4$ across the majority of languages, with some distributions indicating slightly higher confidence, as in the case of Finnish, Indonesian, and Swahili. Lower confidence values can be seen in Korean and Bengali. These observations are in line with the previous findings on performance.

Named Entity Recognition The results of the ensemble NER model are presented in Table 4.2, showing the weighted F1 score across the MasakhaNER 1.0 dataset. Due to hardware limitations at runtime, the *ENG* data is not included. For comparison, the results are shown against the values obtained by Rust et al. (2022). In general, ensemble learning improves the performance significantly for all 9 languages, resulting in scores higher than 90. This is also the case for languages that were previously associated with a low score, such as Amharic (*AMH*). The F1 score gap is 24.3 points in favour of the ensemble method, suggesting that ensemble learning improves the comprehension of long-term dependencies in NER tasks.

5 Discussion

This work showed that it is possible to integrate uncertainty quantification methods and measure calibration in the context of visual text models. These methods include Monte Carlo Dropout at the patch level, with the observation that more work should be directed towards finding more effective ways of aggregating and visualizing uncertainty across longer patch sequences. Attention based methods can also be used to gain insights into how these models encode information, but there remains the debate about whether or not attention counts as an explanation (Bibal et al., 2022). Still, this debate falls outside the scope of this research. Ensemble learning with a low number of individual learners can also be used successfully to improve both performance and confidence.

The results in the MC Uncertainty experiment generally indicate high uncertainty for a high mask ratio. Still, the most optimal value is a mask ratio of 50%, representing a reasonable trade-off between

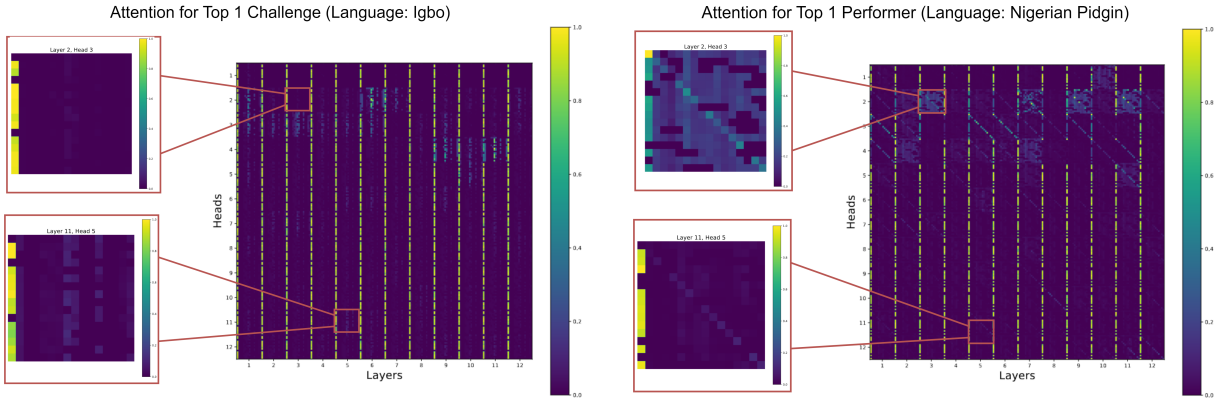


Figure 4.5: Model-level and neuron-level views of attention for the top 1 challenge (left, highest loss value) and performer (right, lowest loss value) in terms of the GNLL loss across all datasets.

	ARA	BEN	FIN	IND	KOR	RUS	SWA	TEL	ENG	**AVG**
PIXEL	57.3	**36.3**	58.3	63.6	26.1	50.5	65.9	63.4	61.7	52.3
Ensemble	**59.5**	35.1	**59.6**	**67.3**	**27.1**	**53.3**	**67.1**	63.4	**62.1**	**54.0**

Table 4.1: The results of the QA task. The ensemble learning model finetuned on the TyDiQA-GoldP dataset is compared with the values reported by (Rust et al., 2022). The metric shown is the F1 score, computed on the `validation` split of the data. The AVG score excludes ENG, as required (Clark et al., 2020).

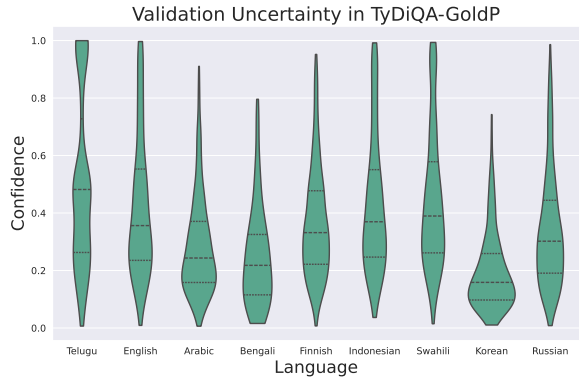


Figure 4.6: Confidence distribution across all languages in the TyDiQA-GoldP dataset for the ensemble model.

uncertainty and loss.

Scripts such as Latin are less uncertain, indicating that multilingual pretraining is necessary. Instead of language, one can focus on introducing a new script, as evidence suggests that there exists knowledge transfer between scripts like Latin and Cyrillic. For example, finetuning on one language such as Chinese might benefit performance in other languages like Korean or Amharic. This approach is more robust than traditional LLMs, where the transfer of learning happens under stricter conditions, for instance when languages share syntactic structures or when there is a significant overlap between vocabularies.

Ensemble learning can be applied successfully to improve performance and calibration in pixel-based language models. The evaluation shows higher F1 scores for 17 of the 19 tested languages across two tasks. The models become more robust and can overcome individual weaknesses by aggregating predictions from multiple learners using hyperparameter tuning. Additionally, ensemble learning improves calibration through better error diversification and data representation.

6 Conclusions and Future Work

The findings of this study indicate that pixel-based language models represent a viable and lightweight solution to traditional language modeling, even for tasks that require semantic understanding of text. Their reliability and explainability can also be improved through uncertainty quantification methods, as shown during the experiments. Future research should focus on perfecting the existing techniques and exploring new ways of understanding the inner workings of models that enccde text as visual representation.

One point to be explored in future works on text reconstruction is the idea of pixels-as-tokens in the context of the Pixel Transformer (PiT) model, introduced by (Nguyen et al., 2024). Instead of training the model to perform patch reconstruction, PiT treats each pixel as a token and the reconstruction happens at the pixel level. Evidence suggests

	AMH	HAU	IBO	KIN	LUG	LUO	PCM	SWA	WOL	YOR	**AVG**
PIXEL	47.7	82.4	79.9	64.2	76.5	66.6	78.7	79.8	59.7	70.7	70.7
Ensemble	**90.2**	**97.1**	**96.1**	**93.9**	**95.5**	**93.1**	**97.1**	**96.1**	**95.8**	**95.2**	**95**

Table 4.2: The results of the NER task. The ensemble learning model finetuned on the MasakhaNER 1.0 dataset is compared with the values reported by (Rust et al., 2022). The metric shown is the F1 score, computed on the `test` split of the data.

that this method completely removes locality as in inductive bias. This can potentially improve long-term context comprehension in the proposed approach, as the current findings indicate that the reconstruction of characters depends on neighboring pixels. Additionally, the finetuning pipeline can be expanded to more complex semantic tasks, such as summarization, open-ended question answering where the answer is not always explicitly mentioned in the context, and text generation (Li et al. (2023) introduced a new method for text generation using GlyphDiffusion). To improve model calibration, post-hoc methods like temperature scaling can be used either separately or in combination with Monte Carlo (Laves et al., 2019). During pretraining, the Cross-Entropy loss can be replaced by the Focal Loss, which is effective in calibration models trained on imbalanced datasets (Wang et al., 2022).

Ethical Considerations

The aim of this study is to shed light on how pixel-based models encode uncertainty. We consider that an explainability analysis should be a prerequisite for any new language model, as this increases users' trust that the technology works as intended and it is not harmful.

In order for this research to exist, we made use of the pretrained PIXEL model provided by Rust et al. (2022). One of the datasets that PIXEL has been pretrained on is the BookCorpus (Zhu et al., 2015) which is well-known for its problematic content and copyright violantions (Bandy and Vincent, 2021). BookCorpus contains books self-published by authors, which did not explicitly consent to including their books in a LLM training dataset, and were not compensated in any way. Moreover, many books contain copyright restrictions which forbid the redistribution of content. Senstive content has also been identified in the data, such as books marked for adult audiences, containing terms and phrases associated with gender discrimination. We acknowledge that by using models trained on problematic data, we risk to further propagate biases. However, these models and datasets are very popular and they cannot be ignored. For this reason, we consider that studying how they work and attempting to explain and interpret them is a goal worth pursuing.

Our paper has a strong focus on language variety, as we explore uncertainty across 18 languages. However, the majority of our fine-tuning data comes from English (as seen in Figure B.1 from Appendix B). This leads to lower performance and less accurate representation in low-resource languages. Once again, this issue boils down to the data available for LLM training, which should ideally be more balanced and representative across diverse linguistic contexts.

Code

We provide the complete implementation for running our experiments on Github, at `https://github.com/stefania-radu/pixel-semantic`.

References

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al. 2021. Masakhaner: Named entity recognition for african languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

David Aldón Mínguez, Marta Ruiz Costa-Jussà, and José Adrián Rodríguez Fonollosa. 2016. Neural machine translation using bitmap fonts. In *Proceedings of the EAMT 2016 Fifth Workshop on Hybrid Approaches to Translation (HyTra)*, pages 1–9.

Jack Bandy and Nicholas Vincent. 2021. Addressing "documentation debt" in machine learning research: A retrospective datasheet for bookcorpus. *Preprint*, arXiv:2105.05241.

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. 2022. Is attention explanation? an introduction to the debate. In *Proceedings of the*

60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3889–3900.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Felix Busch, Lena Hoffmann, Christopher Rueger, Elon HC van Dijk, Rawen Kader, Esteban Ortiz-Prado, Marcus R Makowski, Luca Saba, Martin Hadamitzky, Jakob Nikolas Kather, et al. 2025. Current applications and challenges in large language models for patient care: a systematic review. *Communications Medicine*, 5(1):26.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Falcon Z Dai and Zheng Cai. 2017. Glyph-aware embedding of chinese characters. *arXiv preprint arXiv:1709.00028*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, pages 1–77.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. 2024. Trustllm: Trustworthiness in large language models. *Preprint*, arXiv:2401.05561.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

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

Max-Heinrich Laves, Sontje Ihler, Karl-Philipp Kortmann, and Tobias Ortmaier. 2019. Well-calibrated model uncertainty with temperature scaling for dropout variational inference. *arXiv preprint arXiv:1909.13550*.

Junyi Li, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Renderdiffusion: Text generation as image generation. *arXiv preprint arXiv:2304.12519*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Duy-Kien Nguyen, Mahmoud Assran, Unnat Jain, Martin R Oswald, Cees GM Snoek, and Xinlei Chen. 2024. An image is worth more than 16x16 patches: Exploring transformers on individual pixels. *arXiv preprint arXiv:2406.09415*.

Phillip Rust, Jonas F Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2022. Language modelling with pixels. *arXiv preprint arXiv:2207.06991*.

Elizabeth Salesky, David Etter, and Matt Post. 2021. Robust open-vocabulary translation from visual text representations. *arXiv preprint arXiv:2104.08211*.

Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young. 2018. Super characters: A conversion from sentiment classification to image classification. *arXiv preprint arXiv:1810.07653*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ada Wan. 2021. Fairness in representation for multilingual nlp: Insights from controlled experiments on conditional language modeling. In *International Conference on Learning Representations*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Cheng Wang, Jorge Balazs, György Szarvas, Patrick Ernst, Lahari Poddar, and Pavel Danchenko. 2022. Calibrating imbalanced classifiers with focal loss: An empirical study. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 145–153.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2022. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Limitations

Some limitations of this method include the hardware and training time required to train multiple models. Nevertheless, PIXEL has 20% fewer parameters than BERT, so an ensemble of PIXEL models remains less complex than the BERT variant and significantly more lightweight than models like GPT.

The current study is subject to several limitations. Firstly, the way uncertainty is computed at the image level during the MC experiments can be more reliable. At the moment, uncertainty is averaged across all pixels in an image. However, this does not account for the difference in span length, as some sequences of patches are longer than others. Quantifying uncertainty as an average for each span length in the image could bring more insights into how the model encodes long-term dependencies. Secondly, the information in the attention plots should be aggregated so that all patches are visible at once, while keeping a reasonable image size. Using the current method, visualizing all 256 patches across the 144 attention structures would result in a very large and difficult to interpret image. Regarding the calibration analysis, it is not completely clear that the two measurements of performance (loss vs. MC uncertainty during the pretraining stage and F1 score vs. confidence during finetuning) are quantifying the same underlying metric. For this reason, additional testing should be performed to establish the exact effect size of ensemble learning on model calibration. Moreover, more insights are necessary to establish the trade-off between computational cost, environmental impact and performance gains when training an ensemble of learners compared to a single model.

While it is possible to visualize the attention mechanism in pixel-based language models, there are some comments to be made about this. Unlike traditional language models like BERT where each token represents a meaningful unit and the relationship between two tokens can be understood intuitively, the patches in pixel-based language models cannot be mapped back to text chunks. This makes it more challenging to interpret how attention is paid to the different patches and what are the implications of these connections in the context of the entire model. Moreover, given the large number of attention structures and the image dimensions, visualizing attention for all patches simultaneously becomes very difficult.

B Data Details

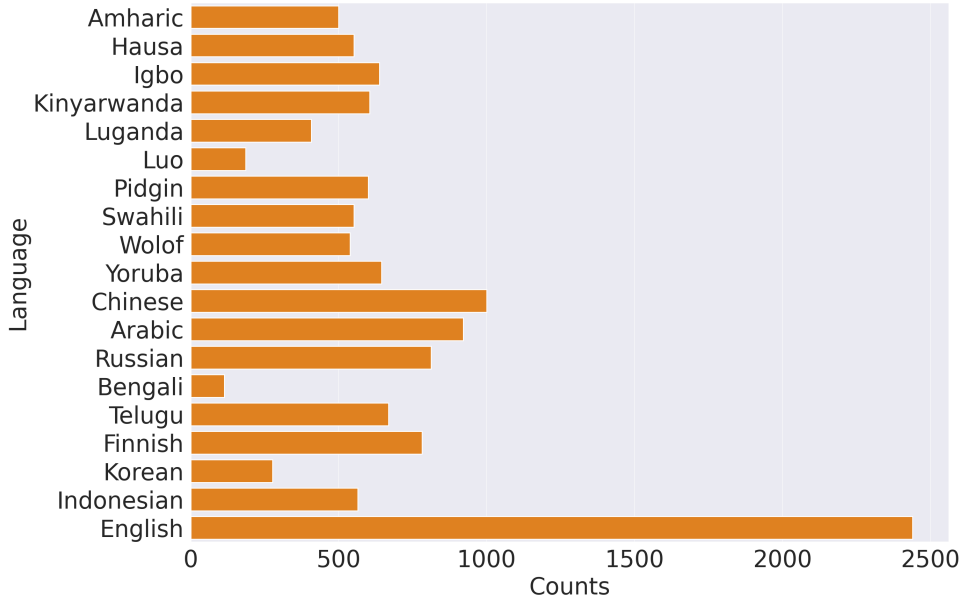


Figure B.1: Distribution of languages used throughout the experiments.

C Experiments Details

Language	ISO 639-3	Language Family	Script
Amharic	AMH	Afro-Asiatic	Ge'ez
Arabic	ARA	Afro-Asiatic	Arabic
Bengali	BEN	Indo-European	Bengali
English	ENG	Indo-European	Latin
Finnish	FIN	Uralic	Latin
Hausa	HAU	Afro-Asiatic	Latin
Igbo	IBO	Niger-Congo	Latin
Indonesian	IND	Austronesian	Latin
Kinyarwanda	KIN	Niger-Congo	Latin
Korean	KOR	Koreanic	Korean
Luganda	LUG	Niger-Congo	Latin
Naija Pidgin	PCM	English Creole	Latin
Russian	RUS	Indo-European	Cyrillic
Swahili	SWA	Niger-Congo	Latin
Telugu	TEL	Dravidian	Telugu
Wolof	WOL	Niger-Congo	Latin
Yorùbá	YOR	Niger-Congo	Latin

Table B.1: An overview of languages used during the experiments. The original PIXEL model is pretrained on English only.

Experiment	Data	Hyperparameters	Metrics
MCU Tasks	NER (MasakhaNER 1.0), SC (GLUE), QA (TyDiQA-GoldP)	$R \in \{0.1, 0.2, \ldots, 0.9\}$, $S \in \{1, 2, \ldots, 6\}$, $W = \{0, 0, \ldots, 0, 1\}, \lvert W\rvert = \lvert S\rvert$	MSE GNLL Uncertainty ($\bar{\sigma}$)
MCU Scripts	Latin, Ge'ez, Chinese Characters, Arabic, Cyrillic, Bengali, Telugu, Korean	$R \in \{0.1, 0.2, \ldots, 0.9\}$, $S \in \{1, 2, \ldots, 6\}$, $W = \{0, 0, \ldots, 0, 1\}, \lvert W\rvert = \lvert S\rvert$	MSE GNLL Uncertainty ($\bar{\sigma}$)
VU	Nigerian Pidgin, Igbo	$R = 0.25$, $S = 6$, $W = \{0.2, 0.4, 0.6, 0.8, 0.9, 1\}$	GNLL Uncertainty ($\bar{\sigma}$)
CA	NER (MasakhaNER 1.0), SC (GLUE), QA (TyDiQA-GoldP)	$R = 0.25$, $S = 6$, $W = \{0.2, 0.4, 0.6, 0.8, 0.9, 1\}$	RMSE Uncertainty ($\bar{\sigma}$)

Table C.1: Overview of the MC Uncertainty experiments. MCU = Monte Carlo Uncertainty; VU = Visualizing Uncertainty; CA = Calibration Analysis.

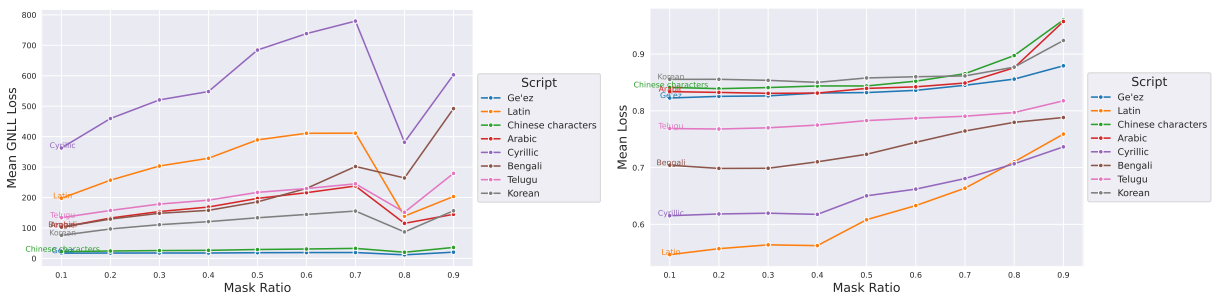


Figure C.1: Mean MSE Loss (left) and GNLL Loss (right) across the different scripts for each mask ratio value R.

Algorithm 1 Patch-level Uncertainty with MC Dropout

Require: Rendered image I, model M, # MC samples $N_{\text{MC}} = 100$, dropout rate $p = 0.1$, patch size $P = 16$
Ensure: Uncertainty map U

Activate dropout in M
for $i \in \{1, \ldots, N\}$ **do**
    $P_i \leftarrow M(I, p)$ ▷ Compute predictions P with dropout
end for
Initialize μ and σ with the shape of I
for each pixel (x, y) **do**
    $\mu(x, y) \leftarrow \frac{1}{N} \sum_{i=1}^{N} P_i(x, y)$
    $\sigma(x, y) \leftarrow \sqrt{\frac{1}{N} \sum_{i=1}^{N} (P_i(x, y) - \mu(x, y))^2}$
end for
Initialize U with the shape of I
for each patch (i, j) in σ **do**
    $\sigma_{\text{patch}} \leftarrow \frac{1}{P^2} \sum_{x=i}^{i+P-1} \sum_{y=j}^{j+P-1} \sigma(x, y)$ ▷ Compute σ per patch
    for $(x, y) \in \{(i, j), \ldots, (i + P - 1, j + P - 1)\}$ **do**
        $U(x, y) \leftarrow \sigma_{\text{patch}}$ ▷ Assign σ_{patch} to all pixels in the patch
    end for
end for
return U

Algorithm 2 Ensemble QA Prediction

Require: k models $\{M_1, M_2, \ldots, M_k\}$, input question q
Ensure: Final answer $\hat{a}$ for the question q

$\mathcal{C} \leftarrow \emptyset$
for each model M_i in $\{M_1, M_2, \ldots, M_k\}$ **do**
    $\mathcal{A}_i \leftarrow M_i(q)$ ▷ Get candidate answers and their confidences
    for each candidate a_j in $\mathcal{A}_i$ **do**
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{a_j\}$
    end for
end for
$\mathcal{C} \leftarrow \left\{ c \mid \sum_{i=1}^{k} \mathbf{1}_{c \in \mathcal{A}_i} = k \right\}$ ▷ Keep the candidates that appear in all models
for each candidate c in $\mathcal{C}$ **do**
    $\text{conf}_c \leftarrow \frac{1}{k} \sum_{i=1}^{k} \text{confidence}_{M_i}(c)$ ▷ Compute average confidence
end for
$\hat{a} \leftarrow \arg\max_{c \in \mathcal{C}} \text{conf}_c$ ▷ Select candidate with highest confidence
return $\hat{a}$

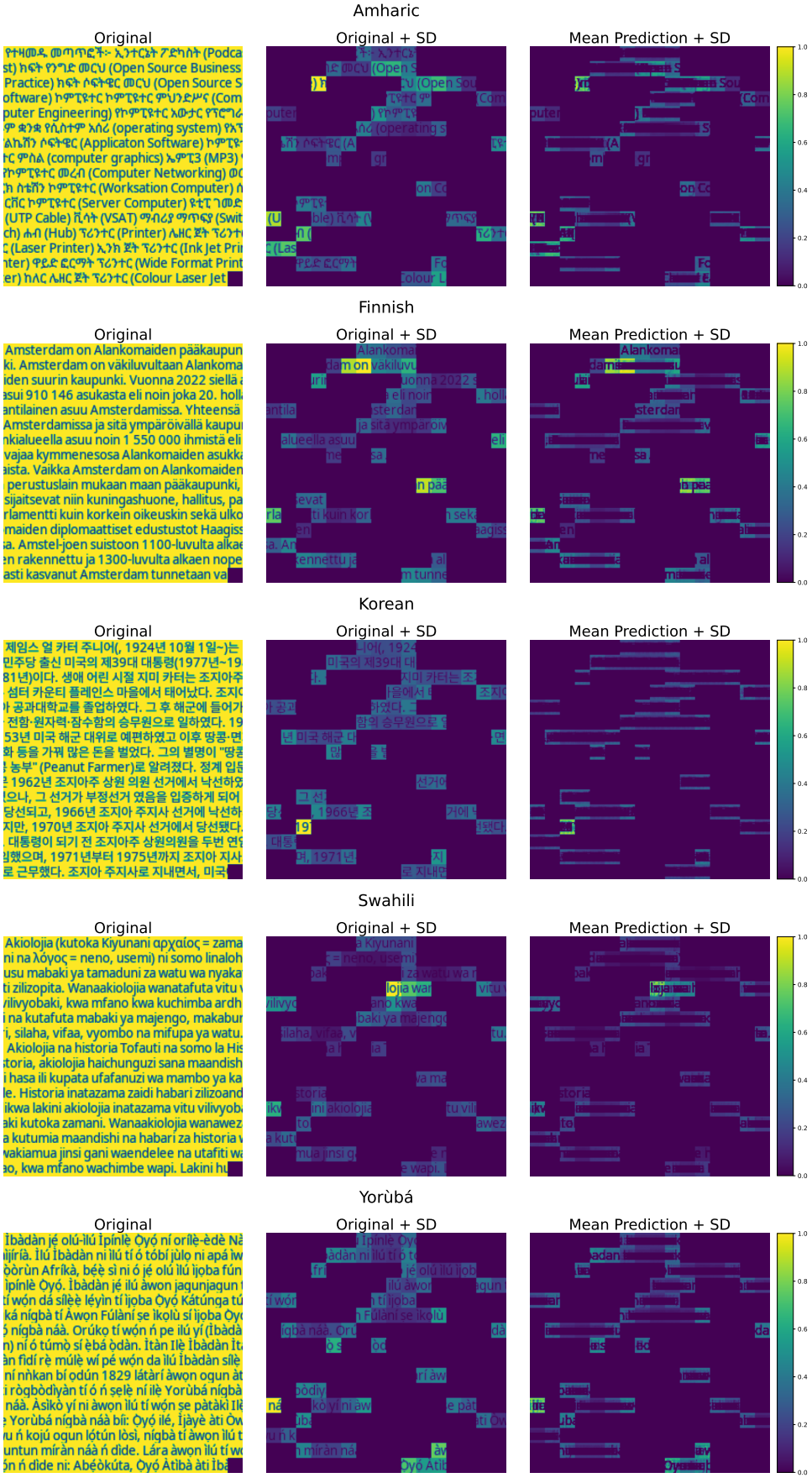


Figure C.2: Examples of uncertainty quantification at the patch-level for various languages.

Parameter	Value
Common Parameters	
Dataset name	tydiqa
Dataset config name	secondary_task
Sequence length	400
Stride	160
Question max length	128
Gradient accumulation steps	1
Max steps	20000
Number of train epochs	10
Early stopping	True
Early stopping patience	5
Evaluation metric	$F1 = \frac{2\times\text{TP}}{2\times\text{TP}+\text{FP}+\text{FN}}$
Doc stride	160
Number of best predictions	20
Model 1	
Batch size	32
Learning rate	7×10^{-4}
Dropout probability	0.15
Seed	101
Model 2	
Batch size	16
Learning rate	7×10^{-5}
Dropout probability	0.15
Seed	102
Model 3	
Batch size	8
Learning rate	7×10^{-5}
Dropout probability	0.05
Seed	103
Model 4	
Batch size	32
Learning rate	7×10^{-6}
Dropout probability	0.1
Seed	104

Table C.2: The finetuning configuration of the QA models, including the common parameters and those changed among the 4 learners.

Parameter	Value
Common Parameters	
Dataset name	masakhane-ner
Sequence length	196
Gradient accumulation steps	1
Max steps	15000
Number of train epochs	10
Early stopping	True
Early stopping patience	5
Evaluation metric	$F1 = \frac{2\times\text{TP}}{2\times\text{TP}+\text{FP}+\text{FN}}$
Model 1	
Batch size	64
Learning rate	5×10^{-5}
Dropout probability	0.1
Seed	100
Model 2	
Batch size	64
Learning rate	5×10^{-6}
Dropout probability	0.2
Seed	101
Model 3	
Batch size	32
Learning rate	5×10^{-5}
Dropout probability	0.1
Seed	102
Model 4	
Batch size	32
Learning rate	5×10^{-6}
Dropout probability	0.1
Seed	103
Model 5	
Batch size	16
Learning rate	5×10^{-5}
Dropout probability	0.2
Seed	104

Table C.3: The finetuning configuration of the NER models, including the common parameters and those changed among the 5 learners.