Bayesian Deep Learning for Natural Language Translation

Submitted in partial fulfillment of the requirements
for the course of MA-308 (Mini Project)
in
3rd Year (6th Semester) of

Master of Science

(Five Year Integrated Program) in Mathematics

Submitted by:

Veer Kamdar (I21MA007)
Abhishek Bisoyi (I21MA013)
Shrihan Pande (I21MA022)

Under the supervision of:

Dr. Sudeep Singh Sanga
Assistant Professor

DEPARTMENT OF MATHEMATICS
SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY
SURAT-395007, GUJARAT, INDIA

April 2024

Department of Mathematics
Sardar Vallabhbhai National Institute of Technology
(An Institute of National Importance, NITSER Act 2007)
Surat-395007, Gujarat, India.

APPROVAL SHEET

Mini Project Report entitled **"Bayesian Deep Learning for Natural Language Translation"** by Shrihan Pande, Abhishek Bisoyi and Veer Kamdar is approved for the completion of the course MA-308 (Mini Project) for the degree of Integrated Master of Science in Mathematics.

Prof. R. K. Maurya
(**Examiner**)

Dr. I. Tripathi
(**Examiner**)

Dr. S. S. Sanga
(**Supervisor**)

Date: 23/04/2024
Place: Surat

Department of Mathematics
Sardar Vallabhbhai National Institute of Technology
(An Institute of National Importance, NITSER Act 2007)
Surat-395007, Gujarat, India.

DECLARATION

We hereby declare that the report entitled "Bayesian Deep Learning for Natural Language Translation" is a genuine record of work carried out by us under the guidance of Dr. S. S. Sanga and no part of this report has been submitted to any University or Institution for the completion of any course.

Shrihan Pande
Admission No.: I21MA022
Department of Mathematics
Sardar Vallabhbhai National Institute of Technology
Surat-395007

Veer Kamdar
Admission No.: I21MA007
Department of Mathematics
Sardar Vallabhbhai National Institute of Technology
Surat-395007

Abhishek Bisoyi
Admission No.: I21MA013
Department of Mathematics
Sardar Vallabhbhai National Institute of Technology
Surat-395007

Date: 23/04/2024
Place: SVNIT, Surat

Department of Mathematics
Sardar Vallabhbhai National Institute of Technology
(An Institute of National Importance, NITSER Act 2007)
Surat-395007, Gujarat, India.

CERTIFICATE

This is to certify that the Mini Project report entitled "Bayesian Deep Learning for Natural Language Translation" submitted by Veer Kamdar, Abhishek Bisoyi and Shrihan Pande in fulfilment for the completion of course of MA 308: Mini Project at Sardar Vallabhbhai National Institute of Technology, Surat is record of their work carried out under my supervision and guidance.

Dr. S. S. Sanga
Assistant Professor
Department of Mathematics
Sardar Vallabhbhai National Institute of Technology
Surat-395007

Date: 23/04/2024
Place: SVNIT, Surat

Acknowledgment

We are very grateful as students of the 5 Years Integrated M.Sc. in Mathematics at Sardar Vallabhbhai National Institute of Technology, Surat. First, we are grateful to our supervisor Dr. S. S. Sanga for his support and guidance in the past 5 months. To Work with him is a great opportunity and has been a pleasure for us. We are grateful to Dr. Anupam Shukla, Director of SVNIT and Dr. Jayesh M. Dhodiya, Head of the Department of Mathematics, as well as all other faculties, research scholars, and non-teaching staff of our department for their regular inspiration and co-activity.

We extend our gratitude to our families and friends for having our back and motivating us, which led to this project being a success.

Veer Kamdar (I21MA007)

Abhishek Bisoyi (I21MA013)

Shrihan Pande (I21MA022)

Abstract

Literary works in any language are exceptionally hard to translate into their complete meaning, this task is still entrusted to professional literary translators who work closely with authors. This is because of the cultural nuances and subtext involved in the composition of any literary work. The advent of Transformer architectures has led to near-100% accurate natural language translation models. However, these models have not been judged by the standard of professional literary translation. Our project aims to improve the translation accuracy of natural language translation models and introduce uncertainty estimation for predictions, through the use of Bayesian Deep Learning. We will compare a non-Bayesian LSTM with a Bayesian LSTM, both employed in a language translation task for this purpose. Since the variables within a Bayesian deep learning model are probability distributions, this provides a number of benefits over non-Bayesian deep learning, these include an inherent ensembling effect, the ability to estimate uncertainty, and increased accuracy even with smaller training data, due to the integration of prior knowledge within the Bayesian framework.

Keywords: Bayesian deep learning · Ensembling · Long Short Term Memory (LSTM) networks · Monte Carlo Dropout · Markov Chain Monte Carlo · Machine learning · Uncertainty Estimation

Veer Kamdar (I21MA007) | Abhishek Bisoyi (I21MA013) | Shrihan Pande (I21MA022)

Contents

Chapter 1

Introduction

In this project, we aim to investigate the usage of Bayesian deep learning in natural language translation tasks. The advantages of using a Bayesian paradigm for inference give us reason to believe that this will improve translation accuracy.

1.1 Scope and Limitations

Scope
By leveraging Bayesian statistical methods, we intend to develop a framework that improves translation accuracy. This framework can be further extended to understand cultural subtleties and contextual interpretations in translation. In the end, our project could improve machine translations, making them more accurate and reliable. This would help bring automated systems closer to the level of human expertise in literary translation.

Limitations
While our framework offers a promising approach to assessing the uncertainty of machine translations, it is not without limitations. Firstly, it may be challenging to capture the full spectrum of cultural nuances and subtext present in literary works, as these aspects often defy precise quantification. Furthermore, our framework may require significant computational resources and expertise to implement effectively. Additionally, while Bayesian methods offer a principled approach to uncertainty estimation, they may not fully capture the complexity of linguistic ambiguity and context dependency inherent in literary translation. Finally, the subjective nature of assessing translation quality may introduce inherent biases into our analysis.

Chapter 2

Literature Review

2.1 Statistical Inference

Given some data $X = (x_1, x_2, ..., x_n)$ from a probability distribution say, $p(x|\theta)$, our goal is to find θ set of parameters that define the probability distribution.

In statistical inference, two major paradigms are frequently used: Frequentist and Bayesian.

Frequentist: This approach views probability as the limit of the relative frequency of an event occurring in repeated trials. It emphasizes properties of estimators and hypothesis tests, such as unbiasedness and consistency.

$$\theta = \text{argmax} p(x|\theta) = \text{argmax} \prod_i p(x_i|\theta) = \text{argmax} \sum_i lnp(x_i|\theta) \tag{2.1}$$

Bayesian: This approach treats probability as a measure of belief or uncertainty. It incorporates prior knowledge and updates it with observed data using Bayes' theorem to obtain a posterior distribution, which summarizes the updated beliefs about the unknown parameters.[5] [10]

$$p(\theta|x) = \frac{\prod_i p(x_i|\theta)p(\theta)}{\int \prod_i p(x_i|\theta)p(\theta)d\theta} \tag{2.2}$$

By playing out the Bayesian paradigm over a large number of samples *(relative to the dimensionality of the outcome)*, the limit of the posterior distribution thus obtained is the same as that of the Frequentist Paradigm.
Hence, we can consider the Frequentist Paradigm to be a special case of the Bayesian Paradigm.

Additionally, the Bayesian Paradigm provides the following benefits:

- There exists an estimation for any number of samples, whose reliability is dependant solely on the prior distribution.
- The prior distribution allows us to encode any prior knowledge we have about the distribution.[10]
- When the Bayes rule is used for training a Neural Network, the prior distribution can act as a form of regularization to counter overfitting.
- A Neural Network that operates within the Bayesian Paradigm allows us to estimate the uncertainty of every prediction. [1]

2.2 Full Bayesian Inference

Let us define a Bayesian probabilistic machine learning model.
x : The set of observed variables.
y : The set of hidden or latent variables.
θ : Model parameters.

ML models can be categorized into two types.

Discriminative: These models cannot generate pairs of observed and hidden variables, they predict the latter from the former.

$$p(y, \theta|x) = p(y|x, \theta)p(\theta) \tag{2.3}$$

Generative: These models can generate pairs of observed and hidden variables.

$$p(x, y, \theta) = p(x, y|\theta)p(\theta) \tag{2.4}$$

We are concerned with Discriminative models for the scope of this project.

2.2.1 The Bayesian Model

Let the training data be (X_{tr}, Y_{tr}) and the discriminative probability distribution be $p(y, \theta|x)$. To train the model, we iteratively improve the prior distribution using Bayes rule. [5]

$$p(\theta|X_{tr}, Y_{tr}) = \frac{p(Y_{tr}|X_{tr}, \theta)p(\theta)}{\int p(Y_{tr}|X_{tr}, \theta)p(\theta)d\theta} \tag{2.5}$$

The output is an ensemble in the form of a distribution, not a single point θ as we obtain in a maximum likelihood estimation (MLE). We perform this training iteratively for some convergence condition.[4]

For testing, or for prediction, we sum our likelihood and prior.

$$p(y|x, X_{tr}, Y_{tr}) = \int p(y|x, \theta)p(\theta|X_{tr}, Y_{tr})d\theta \tag{2.6}$$

The problem with integrals however, is that they are computationally expensive, and they may be intractible.

2.2.2 Tractability and Conjugate Distributions

Closed-form solutions: If the solution of an integral can be expressed with a finite number of operations, that is, without iterative computation or approximation, then it is said to be of closed-form.[10]

Tractable Integrals: Integrals which are solvable using known methods, which require a reasonable time and effort, and produce a closed-form solution are known as tractable.

Conjugate Distributions:
In Bayesian Statistics, $p(y)$(prior) and $p(x|y)$(likelihood) are conjugate if and only if the posterior distribution $p(y|x)$ belongs to the same parametric family as $p(y)$.
That is, if $p(y) \in A(\alpha)$, $p(x, y) \in B(y)$ and $p(y|x) \in A(\alpha')$, then

$$p(y|x) = \frac{p(x|y)p(y)}{\int p(x|y)p(y)dy} \tag{2.7}$$

The denominator is tractable since any distribution in A is normalised. All we need to do is to compute α'. The posterior can only be calculated analytically when such a conjugacy holds.

However, conjugate distributions are rare. Hence we often require approximations for the posterior. A simple approximation can be used as follows:
For training.

$$\theta_{MP} = \text{argmax}p(\theta|X_{tr}, Y_{tr}) = \text{argmax}p(Y_{tr}|X_{tr}, \theta)p(\theta) \tag{2.8}$$

For testing.

$$p(y|x, X_{tr}, Y_{tr}) = \int p(y|x, \theta)p(\theta|X_{tr}, Y_{tr})d\theta = p(y|x, \theta_{MP}) \tag{2.9}$$

This is computationally efficient as it does not involve integrals, however there is a loss of information, and we lose our ensembling capability. [5]

2.3 Approximate Inference

Calculating the posterior predictive distribution, $p(y|x_*, D)$, often proves to be intractable due to the non analytic nature of the integral: $p(y|x_*, D) = \int p(y|x_k, w)p(w|D)dw$
For most models including BNN this integral is not analytic. One can use a simple Monte Carlo Approximation: $p(y|x_*, D) \approx \sum_j p(y|x_*, w_j), w_j$

2.3.1 Monte Carlo Markov Chain Sampling

In variational inference, we have, $p(x, \theta) = p(\theta) \prod_i p(\theta|x)$

- Approximation of $q(\theta|\lambda)$ for posterior $p(\theta|x)$
- Then, $log(p(x)) \geq >= ELBO(\lambda) = \sum iE_{(q(\theta|\lambda)log(p(x_i)|\theta))}$
- Maximize ELBO by minimizing KL-divergence [6]

But, $q(\theta|\lambda)$ is not a good way to approximate. For example, if we consider Ising model, which is used to describe change in magnetic properties of ferromagnetic solids depending on temperature and here the phase transition effect results in a drastic magnetization change within narrow temperature interval.

Consider a Lattice, $x_i \epsilon (-1, 1)$ magnetic moment for one atom.
Entropy $E(x) = -\sum_{(i,j)} x_i x_j - \sum_i (h_i x_i)$
Boltzmann Probability Distribution: $p(x) = \frac{1}{z} e^{-E(x)/T}$ where T is the temperature and z is the normalizing constant. We require the following:

- Mean Energy: $E_{p(x)}E(x)$
- Variance: $D_{p(x)}E(x)$
- Mean Magnetization: $\sqrt{E_(p(x))}\mu^2$ where $\mu = \frac{1}{n} \sum_i x_i$

So, the problem here is that there are 2^n configurations which makes calculating z very hard.

Variational Inference:

- Approximate $p(x) \approx q(x) = \prod_i q_i(x_i)$
- $q_i(x - i)$ = Distribution single atom based solely on neighbourhood
- $q_i(x_i = 1) = \dfrac{1}{1 + e^{(-2/T(h_i + \sum_j E_{qj} x_j))}}$

Since, the interval is small so it lacks precision and in turn, makes the Ising model unfit for use.

In Monte Carlo Markov Chain, $p(x) = \frac{1}{z}\overline{p}(x)$, $z = \int \overline{p}(x)dx$
To construct samples $x_1, ..., x_m$ from p(x) using only $\overline{p}(x)$

$E_{p(x)}f(x) = \int f(x)p(x)dx \approx \frac{1}{m}\sum_i^m f(x_i)$

$x_1 \leftarrow p_o(x)$
$x_2 \leftarrow q(x|x_1)$
$x_3 \leftarrow q(x|x_2)$
...
Therefore, $x_1, ..., x_m$ are not independent but can still be used for $E_{p(x)}f(x)$.

Required properties:

- $\Omega(x)$ invariant under Markov chain if and only if $\int q(x|y)\Omega(y)dy = \Omega(x)$
- Initial distribution = $p_o(x)$, Let $p_i(x)$ be distribution after i step, then Markov chain is ergodic if and only if $p_i(x) \to \Omega(x)$, $i \to \infty$, for all $p_0(x)$ where $\Omega(x)$ is invariant.

2.3.2 Gibbs Sampling

Suppose $x \simeq p(x)$

$x_1^{new} \simeq p(x_1|x_2, x_3, ..., x_n)$
$x_2^{new} \simeq p(x_2|x_1^{new}, x_3, ..., x_n)$
$x_3^{new} \simeq p(x_3|x_1^{new}, x_2^{new}, ..., x_n)$
...
$x_k^{new} \simeq p(x_k|x_1^{new}, x_2^{new}, ..., x_{k-1}^{new})$

Therefore, p(x) is invariant and if all conditions are satisfied then, $p(x_i|x_ı) > 0$ then, it is ergodic.

In Ising Model,

$$p(x_i = 1|x_ı) = \frac{1}{1 + e^{(-2/T(h_i + \sum_j E_{qj}x_j))}} \tag{2.10}$$

Conclusions:

- Applicable for continuous and discrete variables
- Does not have parameters for tuning
- Not good for huge number of parameters

2.3.3 Metropolis Hastings

Algorithm 1 Metropolis-Hastings Algorithm

procedure MetropolisHastings($\pi(\theta), q(\theta'|\theta), N$)
  Initialize θ_0
  for $i = 1$ to N **do**
    Generate candidate sample $\theta' \sim q(\theta'|\theta_{i-1})$
    Calculate acceptance probability $\alpha = \min\left\{1, \frac{\pi(\theta')q(\theta_{i-1}|\theta')}{\pi(\theta_{i-1})q(\theta'|\theta_{i-1})}\right\}$
    Generate $u \sim \text{Uniform}(0, 1)$
    if $u \leq \alpha$ **then**
      Accept candidate: $\theta_i = \theta'$
    else
      Reject candidate: $\theta_i = \theta_{i-1}$
    end if
  end for
  return Samples $\{\theta_1, \theta_2, \ldots, \theta_N\}$
end procedure

2.4 Uncertainty Estimation

Uncertainty is a prevalent and inescapable aspect of deep learning tasks. Deep learning models typically provide a single prediction, which is frequently misinterpreted as a measure of the model's confidence level. However, this prediction is merely a normalized output relative to other possible classes, rather than a true probability of confidence. Consequently, this approach fails to accurately capture the model's overall certainty and can result in problems such as excessive confidence in predictions and unpredictable performance when presented with samples outside the training distribution.[1]

There are two major sources of uncertainty in a Machine Learning model:

- Data uncertainty (Aleatoric Uncertainty) arises from natural randomness in the data, caused by factors like overlapping classes, incorrect labels, and varying levels of noise. This randomness results in predictions with high uncertainty. Increasing the amount of data won't necessarily reduce Aleatoric Uncertainty unless we can observe all factors more accurately. Essentially, Aleatoric Uncertainty is the gap between the information we need and the information we have.[3]
  Aleatoric uncertainty estimator:

$$\frac{1}{T}\sum_{t=1}^{T}\text{diag}(\hat{\sigma}_t^2) \tag{2.11}$$

- Model uncertainty (Epistemic Uncertainty), occurs when a model hasn't learned its parameters well due to limited training data. This results in wide ranges of uncertainty. With only a finite amount of data, it's impossible to precisely determine all of a model's parameters. Essentially, Epistemic Uncertainity represents what the model is unsure about. As more data is used for training, Epistemic Uncertainity typically decreases. It's calculated as the information the model has minus the information it actually shows.
  Epistemic uncertainty estimator:

$$\frac{1}{T}\sum_{t=1}^{T}(\hat{\mu}_t - \mu) \otimes 2 \tag{2.12}$$

In Bayesian deep learning, estimating uncertainty in classification tasks is crucial for robust decision-making. One popular method for uncertainty estimation is Monte Carlo (MC) dropout. In MC dropout, dropout is applied during both training and testing phases, and predictions are sampled multiple times with dropout active to obtain a distribution of predictions. This distribution reflects the model's uncertainty about its predictions.

The uncertainty estimate can be computed using the following equation:

$$\text{Uncertainty} = \sigma_\theta^2(\hat{y}|\hat{x}) + \sigma_{\text{data}}^2(\hat{y}) \tag{2.13}$$

where $\sigma_\theta^2(\hat{y}|\hat{x})$ represents the model uncertainty, which captures uncertainty about the model's parameters, and $\text{Var}_{\text{data}}(\hat{y})$ represents the data uncertainty, which captures uncertainty inherent in the data. By summing these two components, we obtain a comprehensive measure of uncertainty in the model's predictions.

Chapter 3

Methodology

In order to test the advantage in using Bayesian deep learning for language translation, we will compare a non-Bayesian Long Short Term Memory (LSTM) network to a Bayesian LSTM network. Both networks are trained using sequence to sequence modelling.

3.1 Sequence to sequence modelling

Sequence to sequence modelling is a translation model that converts a sequence in one domain to its corresponding sequence in another domain. Within natural language translation tasks, the domains involved are languages and the sequences involved are sentences.[8]

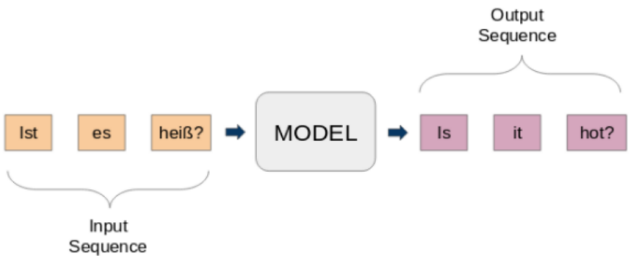


Figure 3.1: Sequence to sequence function

The functioning of both models is as follows:

- We convert a sentence in German to a sequence of Word Embeddings.
- The sequence of Word Embeddings is passed through an encoder LSTM to obtain the latent hidden and cell states.

- These states are then passed on as input to a decoder LSTM to then obtain the corresponding Word Embeddings in English.
- A dense softmax neural network layer converts this last output into discernable words for an English sentence.
- A prediction is obtained for the translation of the German sentence in English.

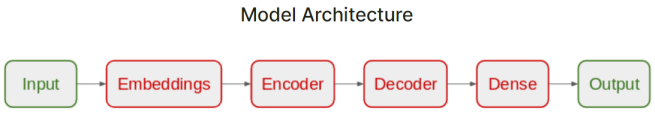


Figure 3.2: Model Architecture

Figures 3.1 and 3.2 depict the functioning of the models.

3.2 Long Short Term Memory

LSTM networks are a variation of recurrent neural networks designed to improve memory retention, addressing the issue of vanishing gradients. They are particularly effective for tasks involving time series data where the duration of time lags is unknown, as they excel at classifying, processing, and predicting such data. There are 3 gates in every LSTM unit as depicted in Figure 3.3 with equations for gate values as given below. The LSTM unit maintains a cell and a hidden state, equations for their update rules involve computation of the aforementioned gate values [9].

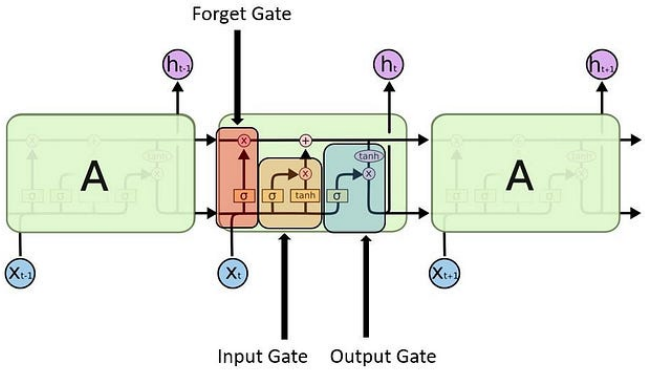


Figure 3.3: An LSTM unit

Input Gate (i_t):

- Description: Controls the extent to which new information flows into the cell state.
- Equation:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{3.1}$$

Forget Gate (f_t):

- Description: Controls the extent to which the information from the previous cell state is retained.
- Equation:

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{3.2}$$

Output Gate (o_t):

- Description: Controls the extent to which the information from the current cell state is used to compute the output.
- Equation:

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{3.3}$$

New Cell State:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3.4}$$

New Hidden State:

$$h_t = o_t \odot \tanh(c_t) \tag{3.5}$$

3.3 Monte Carlo Dropout

We employ Monte Carlo (MC) dropout as a practical approximation to Bayesian inference in our model. MC dropout extends dropout regularization, commonly used during training, to also be used during inference. This extension allows us to estimate model uncertainty without the need for computationally expensive Bayesian inference techniques.[2]

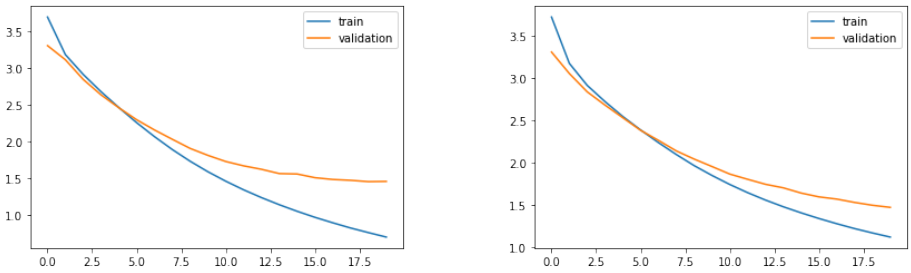
During training, dropout is applied to randomly set a fraction of the neurons to zero, effectively sampling from a subset of the network. This process can be interpreted as performing approximate Bayesian inference over an exponential number of neural network architectures, each corresponding to a different subset of active neurons. [7]

During inference, we extend the dropout technique by performing multiple forward passes through the network with different dropout masks applied. This process effectively samples from the approximate posterior distribution over the network weights. By averaging the predictions from these forward passes, we obtain a more robust estimate of the model's output, along with an estimate of uncertainty in the prediction.

Chapter 4

Conclusion

On building and testing two models, one Bayesian and one non-Bayesian. We have arrived a validation loss of 1.44983 for the non-Bayesian LSTM and 1.47384 for the Bayesian one.



(a) LSTM (b) Bayesian LSTM

Figure 4.1: Cross-entropy loss value

The figures 4.1a and 4.1b depict the decrease in the value of the loss function across epochs during training and validation.

We observe that although Bayesian modelling provides us the benefits of Uncertainty Estimation and is a suitable model for predictions given a small amount of training data, these aspects provide little benefit in natural language translation.
In natural language translation, training data must be large despite the use of Bayesian deep learning. Therefore this advantage is not at play in our model.

Hence we conclude that the use of Bayesian deep learning in LSTM provides no increase in prediction accuracy in natural language translation over regular

deep learning.

4.1 Future scope

The Long Short-Term Memory architecture can be replaced with a Transformer architecture such as the Generative Pre-trained Transformer (GPT) or the Bidirectional Encoder Representations Transformer (BERT). Transformers are better suited to attention based translation tasks such as sequence to sequence, the BERT model has been found to be particularly effective in translation based tasks.

The implementation of any Bayesian transformer will lead to better translations, and it may lead to better insights regarding the contribution of the Bayesian paradigm in natural language translation.

Bibliography

[1] Yarin Gal et al. Uncertainty in deep learning. 2016.

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

[3] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[4] David John Cameron Mackay. *Bayesian methods for adaptive models*. California Institute of Technology, 1992.

[5] Sheldon Ross. Probability and statistics for engineers and scientists. *Elsevier, New Delhi*, 16:32–33, 2009.

[6] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference. *arXiv preprint arXiv:1806.05978*, 2018.

[7] Tao Sun, Bojian Yin, and Sander Bohté. Efficient uncertainty estimation in spiking neural networks via mc-dropout. In *International Conference on Artificial Neural Networks*, pages 393–406. Springer, 2023.

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[10] Alan H Welsh. *Aspects of statistical inference*, volume 246. John Wiley & Sons, 1996.