Spam or Not Spam Data Generation

Ayush Kumar Gupta, Arnav Raj, and Arpit Anil Agrawal

Department of Computer Science and Engineering, Indian Institute of Technology Delhi

November 24, 2024

Abstract

This report outlines a comprehensive methodology for generating synthetic datasets for spam classification using advanced generative models. The problem of dataset insufficiency is addressed by implementing Long Short-Term Memory (LSTM)-based Variational Autoencoders (VAEs). Through detailed preprocessing, model implementation, and evaluation, the project demonstrates the feasibility of creating high-quality synthetic spam and non-spam data, contributing to better spam detection.

1 Introduction

Spam detection has become crucial with the rise of digital communication. Existing datasets are often limited in size and diversity, posing challenges to building robust models. This project addresses this issue by leveraging LSTM VAEs model to generate synthetic spam and non-spam data. Key objectives include generating diverse datasets, ensuring quality, and evaluating models comprehensively.

2 Problem Statement

Effective spam detection relies on diverse and representative datasets. However, existing datasets face several challenges:

- Insufficient size and diversity.
- Privacy concerns when sharing real-world datasets.
- Limited generalization of models trained on static datasets.

This project aims to address these limitations by generating synthetic datasets that mimic the properties of real-world data.

3 Proposed Solution

The project employs:

- **Conditional LSTM-based VAE:** This model generates label-specific text sequences.
- **Evaluation:** Metrics include reconstruction accuracy, KL divergence, and qualitative assessment of generated samples.

4 Dataset and Preprocessing

4.1 Dataset

The "Spam or Not Spam Dataset" from Kaggle was utilized. It contains email text labeled as spam or not spam.

4.2 Preprocessing

Preprocessing involved:

- Lowercasing text for consistency.
- Removing punctuation and non-alphabetic characters.
- Tokenizing text using NLTK.
- Removing frequent and ungrammatical words.
- Creating a vocabulary and mapping tokens to indices.
- Padding sequences to a maximum length of 200.

5 Model Architecture

5.1 Conditional LSTM-based VAE

The architecture comprises:

- **Encoder:** LSTM layers compress input sequences into a latent Gaussian space.
- **Latent Space:** Gaussian distribution with learnable mean and variance.
- **Decoder:** LSTM layers reconstruct sequences from latent space, conditioned on labels.

6 Training and Evaluation

6.1 Training

The Conditional LSTM VAE was trained with the following parameters:

- Batch size: 32
- Epochs: 15
- Optimizer: Adam with learning rate 1×10^{-3}.
- Loss function: Combined reconstruction loss (cross-entropy) and KL divergence.

6.2 Evaluation Metrics

- **Reconstruction Accuracy:** Evaluates the model's ability to recreate input data.
- **KL Divergence:** Assesses the quality of the latent Gaussian space.

  Total Average Loss: 4.4198

7 Dataset and Preprocessing

7.1 Word Frequency Distribution

To analyze the dataset, a word frequency distribution was plotted for the vocabulary. Figure 1 shows the top words in the dataset based on their frequency.

8 Classifier for Testing Generated Data

To evaluate the quality of the synthetic data generated by the Conditional LSTM-based Variational Autoencoder (VAE), a classifier was trained on the original dataset and then used to test the generated data.

8.1 Classifier Training

We used a Support Vector Machine (SVM) classifier for this evaluation. The classifier was trained on the original dataset and validated on a separate validation set. The validation process achieved an accuracy of **99.5%**, indicating that the classifier was effectively trained to distinguish between spam and non-spam data.

8.2 Hyperparameter Optimization

To ensure the SVM classifier was optimally tuned, a Grid Search with Cross-Validation (Grid Search CV) approach was employed. This technique systematically tested different combinations of hyperparameters to identify the best configuration. The optimal parameters obtained were:

- **C:** 10
- **Gamma:** 1

8.3 Testing Synthetic Data

After training, the classifier was used to evaluate the synthetic data generated by the VAE model. This step helped assess the quality and realism of the generated data, as a well-trained classifier should classify synthetic data in a manner consistent with real-world data.

The results of this evaluation further confirmed the utility of the generative model for augmenting spam detection datasets.

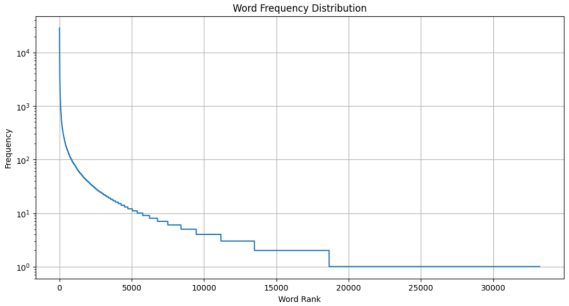

Figure 1: Word Frequency Distribution of the Vocabulary.

9 Training and Evaluation

9.1 Training Loss Progression

During training, the loss values were tracked across epochs. Figure 2 shows the decrease in the total training loss over time, indicating effective learning by the model.

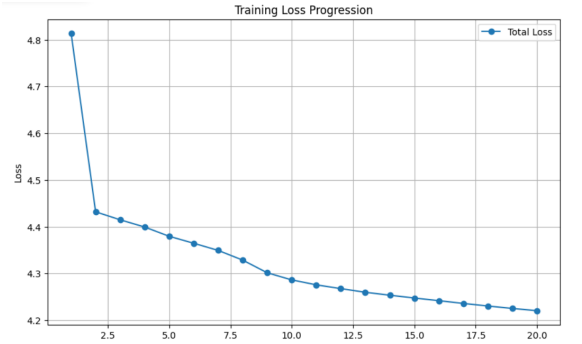

Figure 2: Training Loss Progression of the Conditional LSTM VAE.

10 Results

10.1 LSTM VAE Performance

- Reconstruction loss decreased steadily, indicating effective learning.
- KL divergence values confirmed a well-formed Gaussian latent space.

11 Challenges Faced During the Project

Throughout the project, several challenges were encountered, which required careful consideration and strategic problem-solving:

11.1 Dataset Challenges

- **Limited Size of the Dataset:** The original "Spam or Not Spam" dataset had a restricted number of samples, which constrained the model's ability to learn effectively and generalize well to unseen data.

11.2 Preprocessing Issues

- **Handling Noisy and Ungrammatical Text:** The email texts contained various forms of noise, such as typos, inconsistent formatting, and ungrammatical structures. Cleaning and standardizing this data was essential but time-consuming.
- **Building a Meaningful Vocabulary:** Creating a vocabulary that captures the essential features of the dataset while excluding irrelevant or rare words was challenging. Striking the right balance was crucial for effective model training.

11.3 Model Training Challenges

- **Tuning Hyperparameters for Optimal Performance:** Selecting the appropriate hyperparameters for the Conditional LSTM-based VAE required extensive experimentation. Finding the right combination was essential for achieving optimal performance.
- **Balancing Reconstruction and KL Divergence Losses:** Ensuring that the model effectively reconstructed the input data while maintaining a well-structured latent space involved careful balancing of the reconstruction loss and KL divergence terms in the loss function.

11.4 Synthetic Data Generation Issues

- **Ensuring Diversity and Coherence of Generated Text:** Maintaining both diversity and semantic coherence in the generated synthetic data was a significant challenge. The models needed to produce varied yet meaningful text that accurately reflected spam and non-spam characteristics.

11.5 Learning Curve

As the team was new to LSTM VAEs and Diffusion models, a considerable amount of time was needed to understand and effectively implement these advanced generative models. This steep learning curve required dedicated effort to acquire the necessary knowledge and skills to successfully carry out the project.

12 Conclusion

This project successfully implemented generative models to create synthetic spam and non-spam datasets. The generated data improves training datasets, addresses privacy concerns, and enhances model generalization. Future research could focus on domain-specific model adaptations.

13 Code Repository

The source code for this project is available on GitHub. You can access it using the following link:
GitHub Repository

References

[1] Hakan, O. *Spam or Not Spam Dataset*, Kaggle. Available at: https://www.kaggle.com/datasets/ozlerhakan/spam-or-not-spam-dataset/data

[2] Nic Gian. *Text Generation with VAE*. Available at: https://nicgian.github.io/text-generation-vae/

[3] Author Unknown. *Generating Text with LSTM Networks Using TensorFlow: A Comprehensive Guide*. Available at: https://pub.aimind.so/generating-text-with-lstm-networks-using-tensorflow-a

[4] Veyssier, Laurent. *Text Generation using LSTM*. GitHub Repository. Available at: https://github.com/LaurentVeyssier/Text-generation-using-LSTM

[5] sidharth72. *Text Generation With LSTM*. GitHub Repository. Available at: https://github.com/sidharth72/Text-Generation-With-LSTM