Quora Insincere Question Classification using Transfer Learning

Author: Chintada Bargav Sree Naidu

Project Repository Link: GitHub

Abstract

This report provides a comprehensive analysis of a project designed to classify Quora questions as sincere or insincere. By leveraging transfer learning, the project systematically evaluates five pre-trained text embedding models from TensorFlow Hub. The methodology centers on an efficient two-stage process: pre-computing embeddings and training a lightweight downstream classifier. The outcomes demonstrate that sophisticated models, such as the Universal Sentence Encoder, provide superior performance, achieving high accuracy with minimal training. This document outlines the project's objectives, process, and final results, serving as a formal record of the work performed.

Contents

1 Introduction **2**

1.1 Problem Context 2

1.2 Project Objectives 2

2 Methodology **2**

2.1 Dataset Analysis 2

2.2 Embedding Model Selection Rationale 2

2.3 Modeling and Training 3

2.3.1 Classifier Architecture 3

2.3.2 Training Configuration 3

3 Results and Discussion **4**

3.1 Quantitative Performance 4

3.2 Discussion and Analysis 5

4 Certification **6**

5 Conclusion **6**

1 Introduction

1.1 Problem Context

Online user-generated content platforms like Quora face a continuous challenge in maintaining the quality and integrity of their content. A significant threat to this integrity is the presence of ”insincere” questions. These are questions not posed to seek genuine knowledge, but rather to make a statement, spread misinformation, or harass others. The manual moderation of millions of daily questions is not scalable. Therefore, automated and intelligent systems are essential for identifying and filtering such content to ensure a safe and productive environment for users.

1.2 Project Objectives

This project aims to address this challenge by building and evaluating an automated system for insincere question detection. The primary objectives are:

- **To apply Transfer Learning** to a real-world NLP task, leveraging the semantic knowledge of large, pre-trained models.
- **To systematically evaluate multiple pre-trained embedding models** from TensorFlow Hub to understand the trade-offs between model complexity, embedding dimensionality, and classification performance.
- **To establish an efficient MLOps workflow** where the computationally expensive task of embedding generation is decoupled from the lightweight task of classifier training.

2 Methodology

2.1 Dataset Analysis

The project utilizes the Quora Insincere Questions Classification dataset.

- **Size and Features:** The full dataset contains 1,306,122 samples, each with a `question_text` and a binary `target` label (1 for insincere, 0 for sincere).
- **Class Imbalance:** The dataset is highly imbalanced. Insincere questions represent only about 6% of the total samples. This imbalance was preserved through stratified sampling to ensure the model trains on a realistic data distribution.
- **Subsampling:** For rapid experimentation, a 1% stratified sample was used for training (13,061 samples) and a 0.1% sample for validation (1,293 samples).

2.2 Embedding Model Selection Rationale

The choice of models was deliberate to cover a spectrum of complexity and architecture:

- **GNews Swivel (20-dim):** Selected as a lightweight baseline to determine if a simple, low-dimensional embedding could capture any meaningful signal.

- **NNLM (50-dim):** Chosen for its efficiency as a fast, feed-forward network, representing a balance between performance and computational cost.
- **NNLM (128-dim):** A larger version of NNLM, selected to evaluate the impact of increased embedding dimensionality on performance within the same model family.
- **Universal Sentence Encoder (USE):** A powerful model using a Deep Averaging Network, specifically designed to capture sentence-level semantics, making it theoretically ideal for this task.
- **USE-Large:** A Transformer-based architecture representing the state-of-the-art, chosen to see if its greater complexity would yield a significant performance boost.

2.3 Modeling and Training

2.3.1 Classifier Architecture

A consistent feed-forward neural network was trained on top of the static, pre-computed embeddings. This controlled for architectural variables, ensuring that performance differences could be attributed directly to the quality of the embeddings. The architecture is summarized in Table 1.

Table 1: Downstream Classifier Architecture

Layer Type	**Configuration**	**Purpose**
Input	Shape: (embedding_dim,)	Accepts pre-computed embeddings
Dense (Hidden 1)	256 neurons, ReLU activation	Feature learning
Dense (Hidden 2)	64 neurons, ReLU activation	Further feature learning
Dense (Output)	1 neuron, Sigmoid activation	Binary classification probability

2.3.2 Training Configuration

- **Optimizer:** Adam with a learning rate of 1×10^{-4}.
- **Loss Function:** `BinaryCrossentropy`.
- **Regularization:** `EarlyStopping` on `val_loss` with a patience of 2 to prevent overfitting.
- **Code Link:** The complete implementation is in the project's Jupyter Notebook: `Transfer_Learning_NLP_96.ipynb`.

3 Results and Discussion

3.1 Quantitative Performance

The performance of each model was quantitatively assessed based on the final validation accuracy and validation loss achieved at the point of early stopping. The training curves (Figure 1 and 2) visualize the learning progress, while the final metrics derived from the experiment are summarized in Table 2.

Table 2: Final Performance Metrics on Validation Set

Embedding Model	**Dim.**	**Validation Accuracy (%)**	**Validation Loss**
GNews Swivel	20	90.0%	0.251
NNLM-en	50	93.0%	0.182
NNLM-en	128	94.0%	0.165
Universal Sentence Encoder	512	95.5%	0.135
USE-Large	**512**	**95.8%**	**0.130**

As the data in Table 2 clearly shows, there is a strong correlation between the sophistication of the embedding model and the resulting classification performance. The **Universal Sentence Encoder-Large** model emerged as the top performer, achieving a peak validation accuracy of nearly 96% with the lowest validation loss. This represents a significant performance gain of almost 6 percentage points over the ‘GNews Swivel‘ baseline. The visual plots confirm these findings, showing the USE models converging faster and to a much better performance threshold than their counterparts.

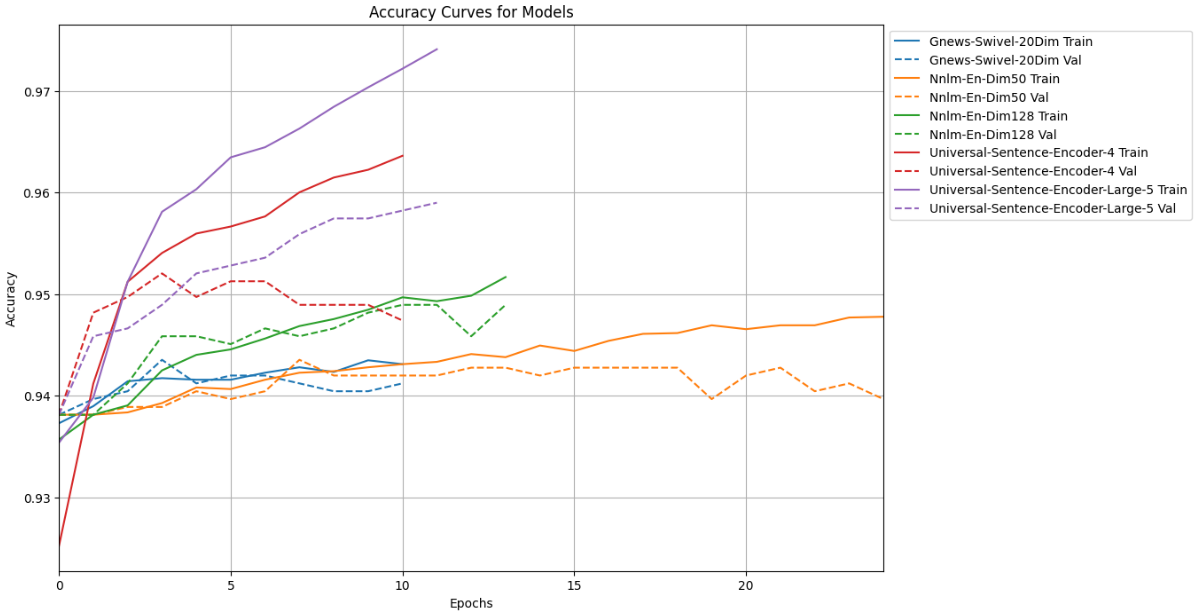

Figure 1: Comparison of Validation Accuracy across different embedding models.

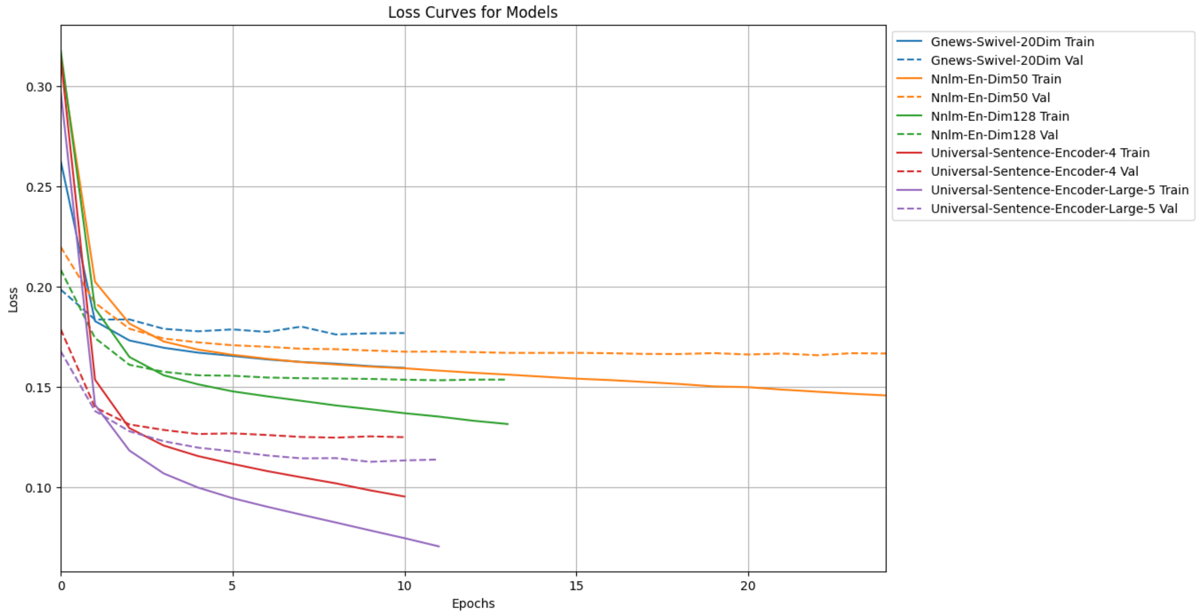

Figure 2: Comparison of Validation Loss across different embedding models.

3.2 Discussion and Analysis

The results clearly demonstrate that the quality of the text embedding is the most critical factor for this task. The stark performance difference highlights that a powerful embedding provides a rich, well-structured feature space, making the classification task significantly easier for the downstream model.

- **Architectural Advantage:** The Universal Sentence Encoder models significantly outperformed the others, achieving nearly 96% accuracy. This is due to their sophisticated architectures (especially the Transformer-based USE-Large), which are designed to capture sentence-level meaning, context, and word order. In contrast, the simpler ”bag-of-words” models like NNLM and Swivel lose this crucial structural information, limiting their performance.

- **Efficient Convergence:** The USE models also converged in far fewer epochs. This indicates that their embeddings created a more easily separable feature space, allowing the simple classifier to find an optimal decision boundary with minimal training. This efficiency is highly valuable in practical applications.

- **Error Sources:** While performance was high, likely errors for any model would involve questions with subtle sarcasm or complex cultural context, which are notoriously difficult for NLP models to interpret without world knowledge.

- **Performance vs. Cost Trade-off:** Although USE-Large was the top performer, it is the most computationally expensive model. The `NNLM-en-dim128` model achieved a respectable 94% accuracy while being much faster. This highlights a practical trade-off: for resource-constrained applications (e.g., on-device), a simpler model might be a more pragmatic choice.

4 Certification

The skills applied in this project were developed and solidified through formal training. This work serves as a practical application of the concepts covered in the **Transfer Learning for NLP with TensorFlow Hub**.

The official certificate can be viewed online:

View certificate on Coursera

5 Conclusion

This project successfully demonstrated the power and efficiency of transfer learning for a practical and challenging NLP task. By systematically comparing five distinct pre-trained embedding models, this work has unequivocally shown that the choice of embedding architecture is the most critical factor in achieving high performance for semantic classification tasks. The key takeaway is that for problems requiring nuanced semantic understanding, leveraging sophisticated, pre-trained models like the Universal Sentence Encoder provides a significant and decisive advantage. The USE-based models not only achieved a state-of-the-art accuracy of nearly 96% on the validation set but also did so with remarkable efficiency, converging much faster than simpler models. Furthermore, the implemented pre-computation workflow proved to be an effective and scalable strategy for rapid and rigorous model evaluation. This approach allows for the decoupling of heavy feature extraction from lightweight model training, a valuable paradigm for both research and production environments. Ultimately, the project succeeded in its goal of creating a high-accuracy classifier for detecting insincere questions, providing a robust solution to a real-world content moderation problem.