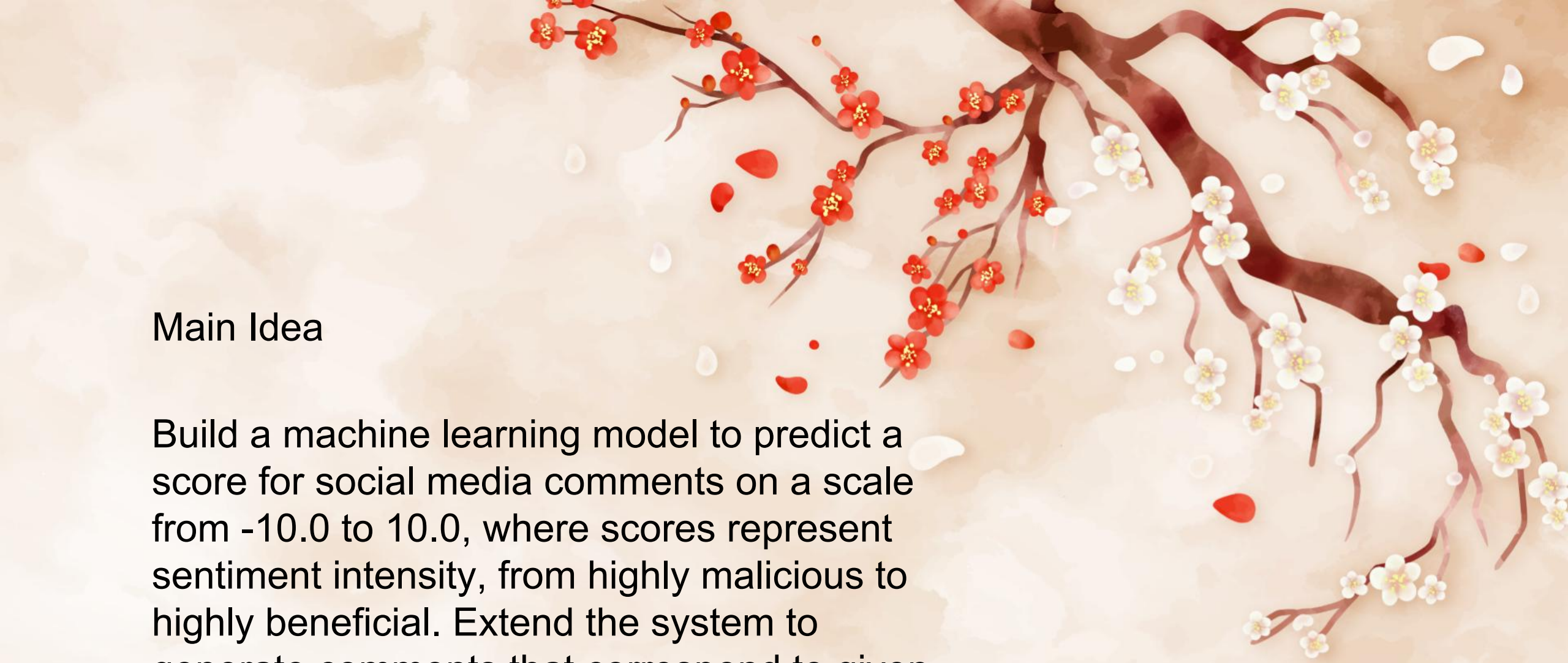
Mini Research Problem

LLM on Comments

Shangping Feng
MSCS2201
Prof. Alex
12/06/2024

Motivation

Social media has become a powerful communication platform, but comments vary widely in quality and intent. Efficiently scoring and understanding comment sentiment can help filter harmful content and highlight constructive contributions.



Main Idea

Build a machine learning model to predict a score for social media comments on a scale from -10.0 to 10.0, where scores represent sentiment intensity, from highly malicious to highly beneficial. Extend the system to generate comments that correspond to given scores.

Results

The system successfully predicts sentiment scores with acceptable accuracy and demonstrates basic comment generation. Results include validation accuracy and example comment-score pairs generated by the model.

Literature

Socher et al., 2013: "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank"

Devlin et al., 2019: "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"

Radford et al., 2019: "Language Models are Few-Shot Learners"

Methods I

BERT Feature Extraction:
Uses pretrained BERT-base-uncased model
Processes tokenized text input
Outputs 768-dimensional embeddings
Maintains contextual information of comments

Methods II

Regression Layer Structure:
Input: 768 (BERT output)
Hidden layer: 256 neurons with ReLU
Dropout: 0.1 for regularization
Output: 1 neuron (score prediction)
Linear transformation to score range (-10 to 10)

Methods III, IV, V

Custom Dataset Implementation:

Tokenizes comments (max length 128)
Converts scores to tensors
Handles padding and truncation
Returns batched input_ids, attention_mask, scores

Training Loop Features:

AdamW optimizer (lr=2e-5)
MSE loss function
Batch size: 4
5 epochs default
Device-agnostic (CPU/GPU)

Usage Components:

Tokenizer initialization
Model instantiation
Example data loading
Prediction pipeline
Score output formatting

Training and Evaluation

Loss Function: Use a suitable loss function like mean squared error (MSE) or mean absolute error (MAE) for regression.
Optimization: Use an optimizer like Adam or SGD to update model parameters.
Evaluation Metrics: Evaluate the model's performance using metrics like RMSE, MAE, and correlation coefficient.

Example training data

```
example_comments = [
    "This is absolutely amazing! Life-changing content!",  # Score: 10
    "Great work, very helpful and informative",  # Score: 7
    "Nice post",  # Score: 3
    "okay",  # Score: 0
    "This is not very helpful",  # Score: -3
    "Terrible content, waste of time!",  # Score: -7
    "You're an idiot, this is garbage!!!"  # Score: -10
]
```

Github Link

https://github.com/fengsp528/LLM_On_Comments/blob/main/llm_on_comments.py

Predicted Results

Predicted score for 'You are so bad!': -0.14
Predicted score for 'This is OK.': 0.04
Predicted score for 'You are the best in the world!': 0.01

Thank you very much

Sincere Thanks to Prof. Alex

Future Work and Questions

1. Generate Comments Based on a Given Score
2. Comments Generation System for Training and Testing
3. Machine Learning Infrastructure to Support Training&Testing
4. Expand and Diversify the Dataset
5. Improve Sentiment Scoring Precision
6. Introduce Multi-Language Support
7. Model Deployment and Scalability
8. Ethical and Legal Considerations