7327873628
pradipta.ss2004@gmail.com

Intent Detection Assignment

Pradipta Sundar Sahoo

IIT Roorkee

Kindly use the Openai Model

2 Deployed project link

Github Link

Deployed in AWS EC2

Amazon EC2

1

a. Framing the Problem as a Machine Learning Task

- Type: Supervised learning (classification).
- Input: Tokenized sentences.
- Output: Class labels (categorical).
- Model: BERT (Encoder-only) and OpenAI models for classification.
- Process: Preprocessing → Tokenization → Fine-tuning model → Classification output.

b. Pros/Cons of Formulations

Encoder-only (BERT)

- Pros:
  - Pretrained model: Leverages transfer learning, improving performance on classification tasks.
  - State-of-the-art performance in sentence-level understanding.
  - Efficient: Handles context and semantics well, making it ideal for classification tasks.
- Cons:
  - High computational cost due to large model size.
  - Primarily focused on classification, not generation.

Decoder-based (OpenAI)

- Pros:
  - Generative: Works well for tasks requiring natural language generation (e.g., conversation).
  - Flexibility: Suitable for multi-tasking and handling diverse NLP tasks.
- Cons:
  - Lower accuracy on classification tasks: OpenAI models are decoder-based and optimized for generating text rather than encoding and classifying it.
  - Inefficiency: For simple classification, decoder models can be overkill, leading to less accurate results due to their generative nature.

Why Encoder is Better for Classification

- Encoder-only models like BERT are specifically designed to understand and encode input sequences, producing context-rich embeddings that help classify text effectively.
- Decoder-based models like OpenAI's GPT focus on generating text and may not be as optimized for discriminative tasks like classification, often leading to lower accuracy in such tasks.
- Encoders, by design, capture the relationships between words within a sentence, whereas decoders are better suited for sequence generation, which doesn't directly benefit classification tasks.

3

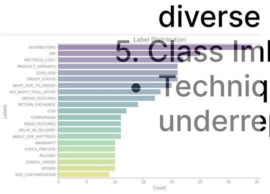
a. Why Do You Think Your Results Make Sense?

The results make sense for the following reasons:

1. Model Choice: Using BERT, an encoder-only architecture, aligns well with the problem's classification nature.
2. Pretrained Model Advantage: Leveraging a pretrained BERT model allows the model to benefit from language understanding learned on large corpora. This is crucial in text classification tasks, as the model doesn't need to learn language nuances from scratch.

b. How Can You Improve Your Model?

1. Data Augmentation:
  - Why: Limited data can restrict the model's ability to generalize well. By augmenting the dataset with techniques like random masking (randomly removing words and requiring the model to predict them), the model can learn more robust patterns and handle a variety of sentence structures.
  - Expected Improvement: This can improve the model's robustness, reducing overfitting and enhancing generalization, especially for low-resource classes.
2. Hyperparameter Tuning:
  - Why: The learning rate, batch size, and epoch count can significantly affect model performance. HPO methods like grid search or random search can help find the optimal settings for better convergence.
3. Fine-tuning Pretrained Models on Domain-Specific Data:
4. Model Ensemble:
  - Why: Combining predictions from multiple models (e.g., combining BERT with other classifiers or multiple BERT variants) can improve results by reducing variance and capturing different aspects of the data.
  - Expected Improvement: Ensembles generally improve accuracy and robustness by aggregating diverse model strengths.
5. Class Imbalance Handling:
  - Techniques like class weighting or over-sampling minority classes could help the model learn from underrepresented classes.



Evaluation

BERT

accuracy			0.89	66
macro avg	0.85	0.89	0.85	66
weighted avg	0.91	0.89	0.89	66

RobertaBERT

accuracy			0.89	66
macro avg	0.83	0.87	0.83	66
weighted avg	0.91	0.89	0.89	66

DistilBert

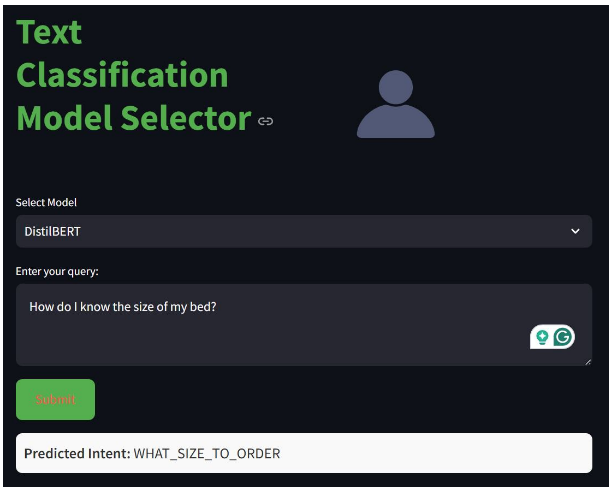
accuracy			0.88	66
macro avg	0.85	0.89	0.85	66
weighted avg	0.88	0.88	0.87	66

OpenAI

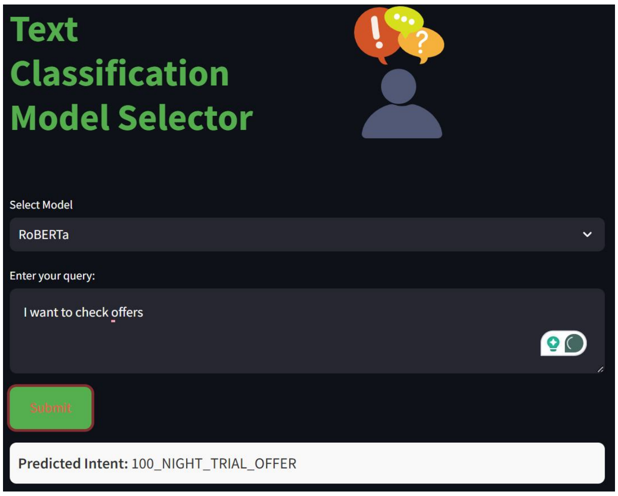
accuracy			0.85	66
macro avg	0.86	0.87	0.82	66
weighted avg	0.92	0.85	0.85	66

Examples

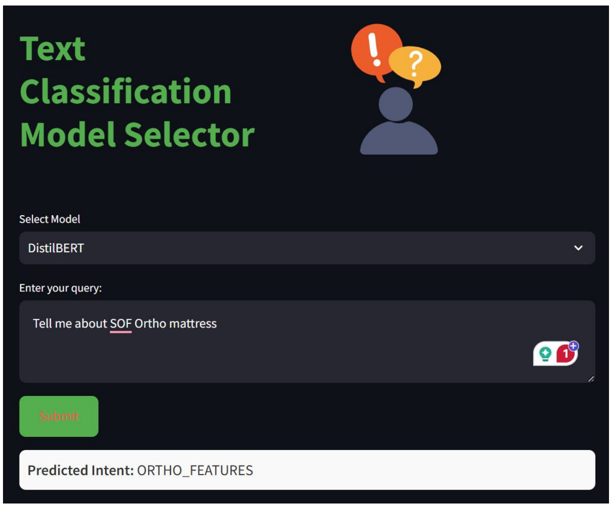
BERT



RoBERTa



DistilBERT



OpenAI