INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY
NEW DELHI

Department of Computer Science & Engineering

CSE 556 : Natural Language Processing (NLP)

Prof. Shad Akhtar

Baseline Results

Anant Kumar Kaushal (2022067)
Anikait Agrawal (2022072)
Ansh Varshney (2022083)

Procedure Description:

1. Seed Initialization:
   - set_seed(seed=42) sets seeds across Python's random, NumPy, and PyTorch libraries for reproducibility.
2. Dataset Loading:
   - Training and validation datasets are loaded from CSV files (train.csv, validation.csv) into Pandas DataFrames.
3. Dataset Conversion:
   - DataFrames are converted into Hugging Face Datasets (Dataset.from_pandas) for compatibility with Hugging Face Transformers.
4. Preprocessing Function (preprocess_function()):
   - Combines the intent (csType) and hate speech (hatespeech) into a single formatted input string: "intent: [INTENT] hatespeech: [HATESPEECH]".
   - Tokenizes the combined input and target counterspeech using BART tokenizer (BartTokenizer.from_pretrained('facebook/bart-base')), with truncation and padding set to a maximum length of 256 tokens.

- Labels (target counterspeech) are tokenized separately using tokenizer's target tokenizer context (tokenizer.as_target_tokenizer()).

5. Dataset Formatting:
    - Preprocessed datasets are formatted into PyTorch tensors (set_format) for input_ids, attention_mask, and labels.
6. Model Initialization:
    - Loads pre-trained BART model (facebook/bart-base) for sequence-to-sequence generation.
    - Initializes DataCollatorForSeq2Seq for dynamic padding during batch processing.
7. Metrics Setup (compute_metrics()):
    - Computes BLEU score (lexical overlap) and BERTScore (semantic similarity) using Hugging Face's evaluate library.
    - Decodes model predictions and labels into readable strings for metric computation.
8. Training Configuration (Seq2SeqTrainingArguments):
    - Defines hyperparameters such as learning rate (5e-5), number of epochs (3), batch size (8), and evaluation strategy (epoch).

9. Model Training (Seq2SeqTrainer.train()):
    - Initiates training using Hugging Face's Seq2SeqTrainer, which automatically handles training loops, evaluation, and checkpointing.
10. Model Evaluation:
    - Evaluates trained model on validation dataset and outputs evaluation metrics (BLEU and BERTScore).

Epoch	Training Loss	Validation Loss	Bleu	Bertscore F1
1	0.366300	0.312772	0.019733	0.861040
2	0.310000	0.296851	0.021457	0.864379
3	0.286700	0.291956	0.022827	0.866201

[3576/3576 42:17, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Bleu	Bertscore F1
1	0.366300	0.312772	0.019733	0.861040
2	0.310000	0.296851	0.021457	0.864379
3	0.286700	0.291956	0.022827	0.866201

[184/184 01:50]

```
=== Validation Evaluation Results ===
{'eval_loss': 0.2919559180736542, 'eval_bleu': 0.022827409129051245,
```

```
'eval_bertscore_f1': 0.8662013249737875, 'eval_runtime': 129.0097, 'eval_samples_per_second': 11.394,
```

```
'eval_steps_per_second': 1.426, 'epoch': 3.0}
```

We can conclude from the results that the eval_bertscore is satisfactory but the eval_bleu score needs to improve.

Possible reasons for this:

1. Inherent Subjectivity: Multiple valid responses reduce exact token-level matches, inherently lowering BLEU scores.
2. Lexical Diversity: High semantic similarity despite varying word choices leads to lower lexical overlap, negatively impacting BLEU scores.

Possible solutions (will be implemented):

1. Two-stage Model Architecture (Baseline 2):

- First stage: Train a lexical reconstruction module explicitly aimed at token-level accuracy.
- Second stage: Fine-tune the model for intent-conditioned generation to preserve semantic coherence.

2. Use of Reinforcement Learning or Fine-tuning with BLEU as a Reward:

- Introduce BLEU-oriented training through reinforcement learning (e.g., Self-critical Sequence Training) to directly optimize token overlap.