README - Topic Spotting with BiLSTM + Self-Attention

This repository provides an implementation of a Topic Spotting (Intent Detection) model for conversational dialogue systems. The approach is based on a BiLSTM network enhanced with a self-attention mechanism. It enables supervised training, evaluation, and optional topic recommendation using a Bag-of-Topics (BoT) representation.

# Requirements

Before running the code, please ensure that the following dependencies and tools are installed:

* Python 3.8 or higher
* pip (Python package manager)
* GPU support recommended (NVIDIA CUDA/cuDNN if training on GPU)

# Installation

1. Clone the repository:

git clone https://github.com/yourusername/topic-spotting-bilstm.git

cd topic-spotting-bilstm

2. Install dependencies:

pip install -r requirements.txt

# Python Libraries

* numpy
* pandas
* scikit-learn
* tensorflow (>=2.8)
* keras
* argparse

# Data Format

The input dataset must be a CSV file with the following columns:  
- utterance: the dialogue text  
- intent: the ground truth label for the utterance  
- split (optional): one of {train, val, test} for pre-defined splits  
  
If no split column is provided, the script automatically splits the dataset into 80% training, 10% validation, and 10% testing.

# Running the Code

Basic usage (with random embeddings):

python train\_topic\_spotting\_from\_csv.py --csv path/to/data.csv

Using pre-trained GloVe embeddings (recommended):

python train\_topic\_spotting\_from\_csv.py --csv path/to/data.csv --glove path/to/glove.6B.300d.txt

With Bag-of-Topics JSON for topic recommendation:

python train\_topic\_spotting\_from\_csv.py --csv path/to/data.csv --glove path/to/glove.6B.300d.txt --bot path/to/bag\_of\_topics.json

# Outputs

* best\_model.h5: trained model weights (best checkpoint)
* final\_weights.h5: final model weights after training
* test\_metrics.json: overall accuracy, precision, recall, and F1-score
* classification\_report.csv: detailed per-class performance report
* vocab.json: tokenizer vocabulary
* label\_classes.csv: label to index mapping
* bot\_recommendations.csv (if BoT provided): utterance-level topic recommendations

# Notes

1. GPU acceleration is strongly recommended for large datasets.  
2. Hyperparameters (learning rate, batch size, max sequence length, etc.) can be configured at the top of the script.  
3. For best results, use GloVe embeddings (glove.6B.300d.txt or glove.840B.300d.txt).