Project Proposal

EE 541: A Computational Introduction to Deep Learning

Project Title: Intent Classification for Natural Language Queries

Group Members: Harry Kang, Lin Lin Hua

Selected Topic: Intent Classification

Problem Description

This project addresses the task of intent classification, a multi-class text classification problem. The objective is to **build a deep learning system that accurately maps a user's natural language query** (e.g., "show me flights to Boston") **to one of 21 pre-defined intent categories** (e.g., *flight* or *airfare*). This capability is a critical component of task-oriented dialogue systems, chatbots, and virtual assistants. The primary challenges lie in building a robust model that can handle variations in phrasing and syntax while also distinguishing between intents that are semantically similar but functionally distinct, such as *flight* versus *flight_time*.

Dataset Analysis

Our work will utilize the **ATIS** (Airline Travel Information System) dataset provided for this course. This dataset consists of approximately 5,800 training utterances and 900 test utterances, distributed across 21 travel-related intent classes. From our initial analysis of the problem description, we have identified two key challenges as below:

1. The dataset exhibits **class imbalance**, where common intents like *flight* appear far more frequently than others. This implies that accuracy alone is a misleading metric. Therefore, we will prioritize the **Macro F1-Score** for evaluation to ensure the model performs well even on rare classes.

2. The data contains a significant amount of **domain-specific vocabulary** (e.g., airport codes, airline names), which will be a critical consideration when designing our vocabulary and selecting an appropriate embedding strategy.

Literature Survey

Our preliminary literature survey reveals a clear technological evolution for solving intent classification on benchmarks like ATIS. Initial approaches relied on traditional machine learning methods, such as Logistic Regression or Support Vector Machines (SVMs), coupled with engineered features like TF-IDF [1]. The field then progressed to classic deep learning, where **Recurrent Neural Networks (RNNs)**, and specifically **LSTMs**, became the standard for their ability to effectively capture sequential dependencies in text [2]. More recently, state-of-the-art results are achieved by fine-tuning large pre-trained Transformer models, such as BERT, which leverage self-attention mechanisms [3]. Our project will systematically explore the performance progression from the traditional baseline to the classic RNN approach, aligning directly with the concepts covered in EE 541, from **Logistic Regression** to **RNNs**.

Proposed Approach

Our methodology is designed as a **systematic experimental investigation** that applies the techniques learned in EE 541. We will use **PyTorch** as our primary framework.

Our first step is to establish a robust non-deep-learning baseline by implementing a **Logistic Regression** model with *scikit-learn* and TF-IDF features. This baseline will provide the minimum performance standard that our deep learning models must surpass.

Next, we will implement our core deep learning model based on **Recurrent Neural Network (RNN)**, specifically an **LSTM**. This model will consist of an *nn.Embedding* layer, an *nn.LSTM* layer, and a final *nn.Linear* layer for classification. The central experiment of our project will be to compare two embedding strategies for this LSTM: (a) **Learned Embeddings**, where the *nn.Embedding* layer is trained from scratch on the ATIS data, and (b) **Pre-trained Embeddings**, where we initialize the layer with **GloVe** vectors to evaluate the effectiveness of transfer learning.

Finally, based on the best-performing embedding strategy, we will conduct a systematic **hyperparameter and architecture optimization** search, applying principles from **Optimizing Training**. This includes tuning the learning rate, batch size, and adding *Dropout* for regularization, as well as exploring architectural changes like a multi-layer LSTM. For evaluation, all models will be compared on the held-out test set using accuracy and Macro F1-Score. We will also generate **Confusion Matrices** to perform a qualitative analysis of failure modes, specifically for the semantically similar classes identified in the dataset.

Timeline and Division of Work

- [Week 12] 11/17 - 11/23:
  - [LinLin] Complete the PyTorch Dataset and DataLoader classes.
  - [Harry] Implement and evaluate the Logistic Regression Baseline (Experiment 1).
  - [Both] Submit this Project Proposal.

- [Week 13] 11/24 - 11/30:
  - [LinLin] Implement and train the core LSTM model (Experiment 3a: Learned Embeddings).
  - [Harry] Implement the GloVe vector integration (Experiment 3b: Pre-trained Embeddings).
  - [Both] Create a unified evaluation script (F1-score, Confusion Matrix).

- [Week 14] 12/01 - 12/05:
  - [Both] Conduct systematic hyperparameter optimization (Experiment 4).
  - [Both] Aggregate all final results and generate comparison charts.
  - [Both] Jointly analyze the confusion matrices for the final report.

- [Finals Due] 12/05:
  - [Harry] Write the Final Report (Methods, Results) and finalize Source Code + README.
  - [LinLin] Write the Final Report (Intro, Analysis, Conclusion) and complete the Model Card.
  - [Both] Final review and submission of all deliverables.

References

[1] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[2] Hakkani-Tür, D., Tür, G., Celikyilmaz, A., et al. (2m016). Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. *In SLT 2016*.

[3] Chen, Q., Zhu, X., & Ling, Z. (2019). BERT-based approach for intent classification on the ATIS dataset. *In Proceedings of the 4th Workshop on Representation Learning for NLP*.

[4] Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2024). *Dive into Deep Learning*. Cambridge University Press. (Course Textbook)