Assignment 2: Slot Filling and Intent Classification on ATIS and SLURP Datasets

Group 14
Vishnu Sudhan Harisankar (2022A7PS1317H)
Sana Jose (2022A3PS0636H)
Sharwari Pejathaya (2022B3AA0792H)
Kriti Saluja (2022B5A70698H)
Priten Rathore (2022B3AA0690H)

Contents

1 Problem Statement

Intent classification and slot filling are fundamental tasks in task-oriented dialogue systems. The goal of this assignment is to build and evaluate models for both tasks using two benchmark datasets, ATIS [1] and SLURP [2]. We implement models with RNN and LSTM architectures and compare their performance across four scenarios:

1. Independent Slot Filling and Intent Recognition
2. Slot → Intent
3. Intent → Slot
4. Joint Model with Shared Encoder (Multi-Task Learning)

Evaluation metrics include precision, recall, F1-score, and accuracy for both slot filling and intent classification.

2 Dataset Statistics

We use two datasets:

- **ATIS:** Airline Travel Information dataset with utterances labeled with intents and slots.
- **SLURP:** Spoken Language Understanding Resource Package, containing more diverse and noisy task-oriented dialogue utterances.

2.1 Basic Statistics

Dataset	Total Utterances	No. of Intents	No. of Slots
ATIS	5,871	22	478
SLURP	17351	60	55

Table 1: Dataset statistics for ATIS and SLURP.

3 Experimental Setup

3.1 Preprocessing

- Tokenization of utterances.
- Conversion of slots and intents to integer indices.
- Padding sequences to uniform length.
- Train-validation-test split: 80%-10%-10%.

3.2 Model Architectures

We experiment with three architectures based on recurrent neural networks to model the sequential nature of the utterances.

3.2.1 RNN Encoder

A Recurrent Neural Network (RNN) processes an input sequence token-by-token. At each step, it combines the current token's embedding with the hidden state from the previous step, which serves as a memory of past information. The final hidden state is used for intent classification, while each token's hidden state is used for slot filling. However, simple RNNs struggle to capture long-term dependencies due to the vanishing gradient problem.

Hyperparameters

- **Epochs:** 10 (for all experiments)
- **Learning Rates:**
  - Independent Intent Recognition: 0.001
  - Independent Slot Filling: 0.01
  - Slot $\rightarrow$ Intent Model: 0.001
  - Intent $\rightarrow$ Slot Model: 0.01
  - Joint Multi-Task Model: 0.0005

System Configuration: Google Colab (Linux 6.1.123+, Intel Xeon CPU @ 2.20 GHz, 12.7 GB RAM)

3.2.2 LSTM Encoder

The Long Short-Term Memory (LSTM) network is an advanced RNN designed to overcome the vanishing gradient problem. It employs a sophisticated cell structure with input, output, and forget gates. This mechanism allows the model to selectively retain or discard information over long sequences, making it more effective at learning long-range dependencies compared to a simple RNN.

Hyperparameters

- **Batch Size:** 32
- **Embedding Dimension:** 100
- **Hidden Dimension:** 128

- **Learning Rate:** 0.001
- **Optimizer:** Adam
- **Epochs:** 10
- **Metric for ATIS:** Macro-averaged F1-score
- **Evaluation Metrics (SLURP):** Macro-averaged and Weighted F1-scores
- **Loss Function(SLURP):** Class-weighted loss for slot filling

System Configuration: Google Colab (Linux 6.1.123+, Intel Xeon CPU @ 2.20 GHz, 12.7 GB RAM)

3.2.3 Joint Multi-Task Model

This architecture utilizes Multi-Task Learning (MTL) to train both slot filling and intent classification simultaneously. A single shared encoder (RNN or LSTM) processes the input sequence to learn a common representation beneficial for both tasks. This shared encoder is connected to two separate output heads: one for slot prediction at each time step and one for sentence-level intent classification. The model is optimized using a combined loss function:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{slot}} + (1-\alpha)\mathcal{L}_{\text{intent}}$$

where α is a hyperparameter balancing the two tasks.

4 Results

4.1 Independent Slot and Intent Models

Model	Dataset	Precision	Recall	F1	Accuracy
RNN	ATIS	0.8096	0.812	0.791	0.985
LSTM	ATIS	0.745	0.732	0.7386	0.9753
RNN	SLURP	0.773	0.671	0.689	0.927
LSTM	SLURP	0.769	0.740	0.7497	0.6699

Table 2: Independent model performance for Slot Filling.

Model	Dataset	Precision	Recall	F1	Accuracy
RNN	ATIS	0.824	0.874	0.841	0.874
LSTM	ATIS	0.670	0.637	0.6531	0.9485
RNN	SLURP	0.769	0.768	0.755	0.758
LSTM	SLURP	0.790	0.785	0.7873	0.78728

Table 3: Independent model performance for Intent Recognition.

4.2 Slot → Intent and Intent → Slot

Model	Scenario	Slot F1	Intent F1	Slot Acc	Intent Acc
RNN	ATIS	0.7912	0.3797	0.9847	0.8735
LSTM	ATIS	0.7386	0.6508	0.9753	0.9530
RNN	SLURP	0.6887	0.5408	0.9277	0.7572
LSTM	SLURP	0.7497	0.7896	0.6699	0.7888

Table 4: Performance for the Slot→Intent scenario.

Model	Scenario	Slot F1	Intent F1	Slot Acc	Intent Acc
RNN	ATIS	0.6722	0.8412	0.9639	0.8735
LSTM	ATIS	0.7402	0.6531	0.9753	0.9485
RNN	SLURP	0.5041	0.7549	0.8835	0.7576
LSTM	SLURP	0.7163	0.7873	0.6237	0.7872

Table 5: Performance for the Intent→Slot scenario.

4.3 Joint Multi-Task Model (Shared Encoder)

Model	Slot F1	Intent F1	Slot Acc	Intent Acc
RNN Shared	0.6241	0.3691	0.9561	0.8791
LSTM Shared	0.7271	0.6364	0.9749	0.9462

Table 6: Performance of the Joint Multi-Task Model on the ATIS dataset.

Model	Slot F1	Intent F1	Slot Acc	Intent Acc
RNN Shared	0.5090	0.5503	0.8942	0.7677
LSTM Shared	0.7240	0.7768	0.6369	0.7784

Table 7: Performance of the Joint Multi-Task Model on the SLURP dataset.

4.4 Analysis

4.4.1 Independent Slot and Intent Models

From Tables 2 and 3, we observe the following trends:

- **ATIS Dataset:**
  - For slot filling, the RNN outperforms LSTM in both F1 score (0.791 vs 0.7386) and accuracy (0.985 vs 0.9753).
  - For intent recognition, RNN achieves higher F1 (0.841 vs 0.6531), although LSTM attains slightly higher accuracy (0.9485 vs 0.874). This suggests that RNNs capture slot-level dependencies better in this dataset, while LSTMs provide more consistent intent predictions in terms of exact matches.
- **SLURP Dataset:**
  - Slot filling performance is comparable between RNN and LSTM (F1: 0.689 vs 0.7497), though LSTM achieves higher F1 and recall, indicating better handling of longer or more complex sequences.
  - For intent recognition, LSTM slightly outperforms RNN in both F1 (0.7873 vs 0.755) and accuracy (0.78728 vs 0.758), showing that LSTM's gating mechanism helps in modeling intent dependencies.

4.4.2 Slot $\rightarrow$ Intent and Intent $\rightarrow$ Slot

From Tables 4 and 5, we notice the influence of one task on the other:

- **Slot→Intent:**
  - On ATIS, passing slot information to intent prediction improves LSTM intent F1 significantly (0.6508 vs 0.6531 in independent model) but reduces RNN intent F1 (0.3797 vs 0.841). This indicates that naive slot-to-intent transfer can hurt simpler RNNs due to noisy slot representations.
  - On SLURP, LSTM benefits strongly from slot information (intent F1 rises to 0.7896), whereas RNN shows moderate improvement.
- **Intent→Slot:**

- On ATIS, RNN slot F1 drops from 0.791 to 0.6722 when using intent information, highlighting the limitations of simple RNNs in leveraging intent context.
- LSTM maintains or slightly improves slot performance with intent guidance (0.7402 vs 0.7386), showing that LSTM can better integrate cross-task signals.
- Similar trends are seen on SLURP, where LSTM outperforms RNN in both slot and intent F1.

4.4.3 Joint Multi-Task Model (Shared Encoder)

Tables 6 and 7 summarize the joint modeling results:

- **ATIS Dataset:**
  - LSTM Shared Encoder significantly outperforms RNN Shared Encoder in both slot F1 (0.7271 vs 0.6241) and intent F1 (0.6364 vs 0.3691), while maintaining high accuracy.
  - This indicates that LSTMs are more effective in multi-task learning setups, likely due to their ability to retain long-range dependencies across both tasks.
- **SLURP Dataset:**
  - LSTM Shared Encoder again achieves higher F1 scores for both tasks (Slot: 0.7240, Intent: 0.7768) compared to RNN (Slot: 0.5090, Intent: 0.5503).
  - The slot accuracy for the shared LSTM drops to 0.6369, suggesting that the shared representation sometimes sacrifices exact token-level predictions to improve overall task synergy.

5 Conclusions

- LSTM consistently outperforms RNN in both independent and multi-task settings, especially in datasets with longer sequences or more complex slot-intent relationships (SLURP).
- Passing information between tasks can improve performance if the model is sufficiently expressive (e.g., LSTM), but may hurt simpler models like RNNs.
- Multi-task learning with a shared encoder offers a trade-off: it improves joint understanding of slots and intents but can reduce token-level slot accuracy in certain datasets.
- Dataset characteristics influence model choice: ATIS, being simpler, allows RNNs to perform reasonably well, whereas SLURP benefits from the gating mechanisms of LSTM.

6 References

- ATIS Dataset: https://huggingface.co/datasets/tuetschek/atis
- SLURP Dataset: https://github.com/pswietojanski/slurp/tree/master/dataset/slurp