Assignment 3

February 28, 2025

Jou-Yi Lee
`jlee1039@ucsc.edu`
University of California, Santa Cruz

Abstract

In Problem 1, we develop an integer linear programming (ILP) formulation for sequence labeling under the BIO tagging scheme. The ILP model uses binary transition variables to represent shifts between labels, and incorporates constraints to ensure valid sequence paths, including special start and stop tokens. In Problem 2, the ILP framework is extended to dependency parsing. We design an ILP model that guarantees each word (except the designated root) receives exactly one parent, while also enforcing projectivity constraints. Furthermore, we discuss additional constraints for ensuring acyclicity within the dependency tree structure. Finally, Problem 3 involves the implementation of an abstractive summarization system for the CNN/Daily Mail dataset using a sequence-to-sequence model with attention. Enhancements such as pointer-generator modules and copy mechanisms are explored to improve the ROUGE scores over the baseline. Together, these approaches demonstrate the versatility of ILP models and neural methods in addressing complex NLP tasks.

1 Consolidated Explanation for Problem 1

In my ILP formulation for sequence labeling, we enforce three essential types of constraints:

- **Transition Constraints:** At each timestep, exactly one transition is selected.
- **Path Continuity Constraints:** The destination tag of the current timestep must match the source tag of the next, ensuring a continuous sequence.
- **BIO Tagging Constraints:** To adhere to the BIO scheme, we prevent illegal transitions (e.g., `I` cannot follow `O`) and require that the first transition from `START` goes to `B`.

The optimal sequence obtained from the ILP solver is:

`Time step 0:`	`START` $\rightarrow$ `B`
`Time step 1:`	`B` $\rightarrow$ `B`
`Time step 2:`	`B` $\rightarrow$ `I`
`Time step 3:`	`I` $\rightarrow$ `O`
`Time step 4:`	`O` $\rightarrow$ `STOP`

This result confirms that:

- The sequence correctly starts with a transition from `START` to `B`.
- Each step follows continuously with the destination of one timestep becoming the source of the next.
- The BIO constraints are satisfied (no `O` is followed by `I`), ensuring a valid tagging sequence.

2 Problem 2: Dependency Parsing with an ILP

2.1 Key Points

- **Dependency Tree Variables:** I correctly define the binary variables $x[i,j]$ for $i \in \{0,1,\ldots,n\}$ and $j \in \{1,\ldots,n\}$ such that $x[i,j]=1$ if word w_i is the parent of w_j. Self-loops (i.e., $x[j,j]$) are not allowed.
- **One Parent per Non-root Word:** Every non-root word receives exactly one parent.

- **Objective Function:** I maximize the sum of scores over all dependency edges.

2.2 Implementation

- **Sentence and Word Setup:** I set $n = 5$ to represent a sentence with 5 non-root words (i.e., $w_1, \ldots, w_5$), and the word list is defined as

  ```
  ["$","w1","w2","w3","w4","w5"].
  ```

- **ILP Model Initialization:** I initialize the ILP model with the objective of maximization.

- **Decision Variables:** I define binary variables $x_{i,j}$, where i ranges from 0 to n (with $i = 0$ corresponding to the root) and j ranges from 1 to n.

- **Constraints:**

  1. **No Self-loops:** For each non-root word w_j, enforce $x_{j,j} = 0$.
  2. **Exactly One Parent per Non-root Word:** For every w_j (with $j \geq 1$), impose
  $$\sum_{i=0}^{n} x_{i,j} = 1.$$
  3. **Optional Projectivity Constraint:** For each potential edge from w_i to w_j (with $i < j$), if there is a span (i.e., words exist between w_i and w_j), then the number of words between them, given by $j - i - 1$, multiplied by $x_{i,j}$ is bounded by the sum of dependency assignments for words within that span.

- **Score Definition:** A score for each potential dependency edge, $score(i, j)$, is generated using a random value between -1 and 1.

- **Maximization Objective:** The ILP model is set to maximize the total score, ensuring that the solution corresponds to the highest-scoring dependency tree.

2.3 Result Explanation

Dependency Parsing Tree:

```
Parent of w1 is $
Parent of w2 is $
Parent of w3 is $
Parent of w4 is w5
Parent of w5 is w3
```

1. w_1, w_2, and w_3 are directly attached to the root ($).

2. w_4 is dependent on w_5, and w_5 is dependent on w_3.

3. All non-root nodes have exactly one parent, and no self-loops occur. The optional projectivity constraint (if active) ensures that the structure adheres to projectivity rules.

3 Problem 3: Summarization System for the CNN/Daily Mail Dataset

3.1 Really Sorry for the Delay !!!!!!!