Assignment 3
NLP 203: Natural Language Processing III

University of California Santa Cruz

All assignments are to be completed individually. You may discuss with the TAs and the instructor, but you may not receive help from anyone else.

Instructions. Submit a zip or tgz file containing your writeup, and output and code for problem 3 to Canvas. Although not required, we encourage you to use LaTeXto typeset your writeup.

Problem 1 [15 points]

A linear program (LP) in standard form looks like this:

$$\max_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \text{ subject to } \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$$

Let N be the dimensionality of x and let M be the number of linear constraints (not including the positivity constraints); A $\in R^{M\times N}$ and b$\in R^M$. Any linear program can be transformed to this form.

An *integer linear program* (ILP) adds the constraint that each x_i is integer-valued. Often bounds constraints and integer constraints are merged into a constraint that each $x_i \in 0, 1$.

In this problem, we will be creating an ILP for sequence labeling.

1. For each timestep $t = 0 \ldots T$, have a binary random variable x_{ij}^t for each possible transition from tag i to tag j. Write the objective function (the function we want to maximize) as a linear function of x. Include transitions from the start and stop tags.

2. Express the constraints that there must be exactly one transition at each timestep as a set of linear inequalities ($\leq$). Hint: to express an equality constraint, you can use two $\leq$ inequality constraints.

3. The transitions must form a valid path, so that a transition from i to j must be followed by a transition from j to k. Express this a inequality constraints. We now have a complete ILP for sequence labeling.

4. Imagine we are doing BIO tagging, and want to enforce the constraint that tag I cannot follow tag O. How would you do this with additional inequality constraints?

Problem 2 [15 points]

Dependency parsing with an ILP.

1. Recall the standard form for an ILP is

$$\max_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \text{ subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$$

where x $\in R^N$, A $\in R^{M\times N}$, b$\in R^M$, and c $\in R^N$.

Let $w = \langle w_0 = \$, w_1, w_2, \ldots, w_n\rangle$ be a sentence. ($ is the root or "wall" symbol.) Let $score(w_i, w_j)$ be the score associated with making w_i the parent of w_j in a dependency tree and let the score of a dependency tree be

$$\sum_j score(w_{\text{parent}(j)}, w_j)$$

Your job is to define an ILP such that minimizing x can be converted back into a dependency tree such that:

- every word in $\langle w_1, \ldots, w_n\rangle$ has exactly one parent in $\{w_0, w_1, \ldots, w_n\}$;
- w_0 does not have a parent
- the solution to the ILP will correspond to the highest scoring dependency tree.

Note: for this part, you do not need to include any additional constraints beyond these mentioned.

To answer the question, you must define x (in terms of the dependency tree), A $\in R^{M\times N}$, b $\in R^M$, and c $\in R^N$ for an ILP in standard form given in problem 1. Hint: you should be able to define a solution in which $N = O(n^2)$ and $M = O(n)$. You don't need to convert to standard form, and in this problem it's okay to use equality and inequality constraints as long as they are linear.

2. Do part 1 again, but this time add and overall **projectivity** constraint on the tree: if w_i is the parent of w_j, then $\forall k \in (i, j) \cup (j, i)$, the parent of w_k is also $\in [i, j] \cup [j, i]$. (Another equivalent statement is that for any i, w i and all its descendants for a contiguous substring.) Hint: this will require at least another $O(n^2)$ linear constraints. Note: do not worry about cyclicity yet, unless you find it helpful to do so.

3. **Bonus question:** The dependency structures recovered above have not constrained to be acyclic. In practice we typically want dependency structures to be trees. Can you define constraints that will enforce acyclicity? You may introduce more variables, too, if you need to. You may also use non-linear constraints.

Problem 3 [15 points]

In this problem, you will need to implement an abstractive summarization system on *CNN/Daily Mail Dataset*, which consists of more than 300K news articles and each of them is paired with several highlights, known as multi-sentence summaries. We have summarized the basic statistics of the dataset in the table. You can download the dataset and starter code from here[1].

[1] https://drive.google.com/drive/folders/1DtmHU9Qf3WWa3cdnTWzehGpKQ-5LVUhH?usp=sharing

	CNN/Daily Mail		
	Train	**Dev**	**Test**
# of article-summary pairs	287,227	13,368	11,490
Average article length	751	769	778
Average summary length	55	61	58

You are asked to implemented a **seq2seq with attention** model. We have provided the starter code of a simple seq2seq with attention for machine translation in the zip file for you to learn. You can use this starter code or write your own code base. Once you get the predicted summaries from your trained model, run the script `evaluate.py` (Note: you need to install the python library rouge[2]). Be sure to set the paths of the gold and your predicted files correctly in the script `evaluate.py` first. You are required to report the F-score of three metrics: **ROUGE-1** (unigram), **ROUGE-2** (bigram) and **ROUGE-L** (longest common subsequence). To get full credit, your results should be above the baseline **0.2, 0.04 and 0.2**, respectively.

Next, improve upon your baseline by trying at least one method to improve the performance, such as adding pointer-generator module, copy mechanism or trying other encoders/decoders, etc. However, to focus your efforts on improvements to the model directly, do not use extra data resources, such as pre-trained models like BERT.

Suggestions You can upload the starter code notebook to Google Colab for your experiments, or run it on the nlp-gpu-01 server. It may take more than 10 hours to train your model to convergence with early-stopping, so be sure to start early. We also suggest debugging your model on a smaller dataset.

Deliverables In the writeup, be sure to fully describe your models, experimental procedure and any changes you made. Report the F-score of the three metrics. Submit your final prediction file and your code in the zipfile containing your writeup.

[2]https://github.com/pltrdy/rouge