Assignment 2

CS 421: Natural Language Processing

Due: March 14, 2025 (11:59 p.m. CST)

1 Introduction

Welcome to Assignment 2 for CS 421! This assignment is designed to give you hands-on experience with one of the most well-known applications of Natural Language Processing (NLP), *Machine Translation*. You will explore both traditional and modern approaches to machine translation, implementing models and evaluating their outputs. Throughout the process, you will gain a deeper understanding of how neural networks handle language generation and translation tasks.

This assignment consists of two main deliverables. First, in the coding component, you will implement **Recurrent Neural Networks (RNNs)** and **Long Short-Term Memory networks (LSTMs)** using PyTorch, as well as work with **Transformers** via the Hugging Face library. These models will help you understand the evolution of machine translation techniques, from sequence-based architectures to the more advanced attention-based approaches. Second, in the written component, you will manually evaluate three translated samples using the **BLEU (Bilingual Evaluation Understudy)** metric to assess translation quality. This will give you insight into how automatic evaluation metrics work and their limitations in real-world applications.

By the end of this assignment, you should have a solid grasp of implementing and evaluating machine translation models. Whether you are new to deep learning-based NLP or already familiar with these techniques, this assignment will provide valuable practice in building and assessing machine translation systems.

Don't hesitate to reach out on Piazza or during office hours with any questions you have as you complete it! Happy Coding!

2 Instructions

Each part of this deliverable is labeled as Code or Written. The guidelines for these two types of components are provided below.

2.1 Code

The questions need to be completed using Python (version 3.10+). There are **no external packages** required to complete this assignment. If you want to use an external package for any reason, you are required to get approval from the course staff on Piazza prior to submission. Starter code is provided for the assignment in the form of a Jupyter notebook in the supplementary material. Do not rename/delete any functions or global variables provided in this python notebook, and write your solution in the specified sections. The python notebook may also contain important information and/or examples in comments, so please read them carefully. This part of the assignment will be graded manually after submission on Gradescope.

To implement and train your models, please utilize Google Colab[1]. Please take note of the usage limits on Google Colab and plan your work accordingly. You may also implement and train your models locally if you have access to enough compute resources (A dedicated GPU with alteast 8GB of VRAM). The assignment is designed to test for correct implementation. Training hyperparameters are selected to ensure that training is possible in a Google Colab environment.

To submit your solution for Code questions, you need to download the `.ipynb` file[2]. You need to submit the following file as a part of your submission:

- □ `translation.ipynb`

Submit this `.ipynb` file on Gradescope under Assignment 2. All specified files need to be submitted to receive full credit.

2.2 Written

You are required to submit all Written questions in a single PDF file. You may create this PDF using Microsoft Word, scans of your handwritten solution, LATEX or any other method or design tool you prefer. To submit your solution for Written questions, you need to provide answers to the following questions in a single PDF.

- □ Q11

Before submission, ensure that all pages of your solution are present and in order. Submit this PDF on Gradescope under Assignment 2. Please match all questions to their respective solutions (pages) on Gradescope. Questions not associated with any pages will be considered blank or missing, and all questions need to be completed to receive full credit.

[1] http://colab.research.google.com/

[2] To download the `.ipynb`, select `Download > Download .ipynb` from the `File` menu on Google Colab

3 Questions

3.1 Code (85 points)

Q1 (5): Prepare Dataset
Follow Q1 in the Jupyter notebook to download the dataset and split it into training, testing, and validation sets.
Supplementary material: `translation.ipynb`

Q2 (10): Prepare for training RNNs
Follow Q2 in the Jupyter notebook to tokenize the dataset, define a custom dataset, and define the dataloaders.
Supplementary material: `translation.ipynb`

Q3 (10): Implementing RNNs
Follow Q3 in the Jupyter notebook to define the RNN model in the `Seq2SeqRNN` class. Add the required layers to the neural network and define the forward pass for the RNN model.
Supplementary material: `translation.ipynb`

Q4 (15): Training RNNs
Follow Q4 in the Jupyter notebook to add code for training the RNN model. You will need to initialize the RNN model, define the hyperparameters, define the loss and optimizer functions, and implement a custom training loop.
Supplementary material: `translation.ipynb`

Q5 (5): Evaluating RNNs for Machine Translation
Follow Q5 in the Jupyter notebook for evaluating the performance of the trained RNN model for the machine translation task. Implement the calculation of BLEU-1,2,3,4 scores using the `sacrebleu` library for the test dataset.
Supplementary material: `translation.ipynb`

Q6 (10): Prepare for training Transformers
Follow Q6 in the Jupyter notebook to tokenize the dataset, process data into transformer inputs, and define a data collator and dataloaders.
Supplementary material: `translation.ipynb`

Q7 (5): Choosing and loading Transformers
Follow Q7 in the Jupyter notebook to choose an appropriate Transformer model from HuggingFace[3]. Load and initialize the pretrained checkpoint for the chosen Transformer model.
Hint: We recommend `google-t5/t5-small` or a similarly sized model if training on Colab.
Supplementary material: `translation.ipynb`

[3] `https://huggingface.co/models`

Q8 (15): Training Huggingface Transformers
Follow Q8 in the Jupyter notebook to add code for training the Transformer model. You will need to initialize the Transformer model, define the hyperparameters, define the loss and optimizer functions, and implement a custom training loop.
Supplementary material: `translation.ipynb`

Q9 (5): Evaluating Transformers for Machine Translation
Follow Q9 in the Jupyter notebook for evaluating the performance of the trained Transformer model for the machine translation task. Implement the calculation of BLEU-1,2,3,4 scores using the `sacrebleu` library for the test dataset.
Supplementary material: `translation.ipynb`

Q10 (5): Inferencing on Transformers
Follow Q10 in the Jupyter notebook for inferencing using the trained Transformer model for the machine translation task. You would be required to implement the inference pipeline, which provides the English translation for an input sentence in French.
Supplementary material: `translation.ipynb`

3.2 Written (15 points)

Q11 (15): Manual BLEU evaluation
In this part, you are required to manually calculate the BLEU score for the translated output of the following sentences from French to English. Calculate the BLEU-1, BLEU-2, BLEU-3, and BLEU-4 scores by comparing the given English translation to the translation provided by your trained Transformer model using the inference pipeline as defined in Q10. Provide steps for your calculation of the BLEU score[4].
French: Chicago est célèbre pour ses pizzas profondes, son jazz et son architecture époustouflante.
English: Chicago is famous for its deep dish pizza, jazz and stunning architecture.
French: J'ai traduit cette phrase du français vers l'anglais.
English: I translated this sentence from French to English.
French: Vous avez maintenant terminé le deuxième devoir de ce cours.
English: You have now completed the second assignment of this course.
Supplementary material: `translation.ipynb`

[4] For reference on how BLEU is calculated: `https://www.geeksforgeeks.org/nlp-bleu-score-for-evaluating-neural-machine-translation-python/`

4 Rubric

This assignment will be graded according to the rubric below. Partial points may be awarded for rubric items at the discretion of the course staff.

Q1 (5 points possible)	
	+5
Q2 (10 points possible)	
	+10
Q3 (10 points possible)	
	+10
Q4 (15 points possible)	
	+15
Q5 (5 points possible)	
	+5
Q6 (10 points possible)	
	+10
Q7 (5 points possible)	
	+5
Q8 (15 points possible)	
	+15
Q9 (5 points possible)	
	+5
Q10 (5 points possible)	
	+5
Q11 (15 points possible)	
BLEU Scores correctly calculated for Sentence 1	+5
BLEU Scores correctly calculated for Sentence 2	+5
BLEU Scores correctly calculated for Sentence 3	+5