Iran University of Science & Technology

School of Computer Engineering

Assignment #5

Natural language processing

BY:

DR. Behrouz Minaei, Fall 2024

Teaching Assistants:

Mahdi Feghhi

Due: 1403/10/10

Contents

Notes

1. Submit the answers in a complete PDF file and the code for the questions in the .ipynb format (including the notebook cell outputs) in a compressed file named HW4_StudentID.zip by the specified deadline.
2. A total of 72 + 72 hours of delay in submitting the answers is allowed across all projects. After that, for each additional day of delay, 10% of the score will be deducted.
3. If a student submits the project earlier than the deadline and achieves 75% of the score, up to 24 hours will be added to their allowable delay time.
4. The maximum delay for submitting each assignment is 5 days, and after 5 days, submission will not be accepted.
5. It is important to note that the explanation of the code and the obtained results must be included in the PDF file. Code without a report will result in a score deduction.
6. The evaluation of the assignment will be based on the correctness of the solution and the completeness and accuracy of the report.
7. Assignments must be completed individually, and group work on assignments is not allowed.
8. Please allocate sufficient time for the assignment and avoid leaving it until the last days.
9. You can ask your questions in the relevant group.

good luck.

Problem 1

Explain the differences between generative classifiers (e.g., Naive Bayes) and discriminative classifiers (e.g., logistic regression) in terms of their modeling approach and computational objectives. Provide an example to illustrate the distinction.

Problem 2

In this problem, you should use the following sentences to implement POS tagging using the HMM method on this sentence: "<S> Can Will hunt Mark? <E>".

- <S> Will mark hunt. <E>
- <S> Mark will hunt. <E>
- <S> Can Will mark? <E>
- <S> Mark can hunt. <E>

Use only "Noun", "Verb", and "Modal" tags. <S> tags the start of the sentence and <E> tags the end of it. Follow the following steps to achieve the required results:

a. First, apply POS tagging on the given sentences and create a table that shows the probability that each word is a Noun, Modal, or Verb. You can use the first row of the following table as an example: **(5 points)**

Word	Noun	Verb	Modal	Total Count
will	2	0	1	3
mark				
hunt				
can				

b. Second, calculate the probability of two labels occurring together. You can use the following table structure: **(5 points)**

	<S>	Noun	Verb	Modal	<E>
<S>					
Noun					
Verb					
Modal					
<E>					

c. Finally, use the results of the previous sections to perform POS tagging on <S> Can Will hunt Mark? <E>. You need to draw a graph of the sentence, then delete the edges that have zero probability to get the answer. **(10 points)**
d. Additionally, use few-shot prompting from a large language model (LLM) to perform POS tagging on the same sentence: <S> Can Will hunt Mark? <E>. Provide a few example sentences along with their POS tags as part of the prompt to guide the model. Compare the results from the LLM with those obtained using the HMM method. **(5 points)**

Problem 3

According to the following grammar and using the CKY algorithm, perform the parsing operation by drawing a table for the following sentence:

Grammar Rules:

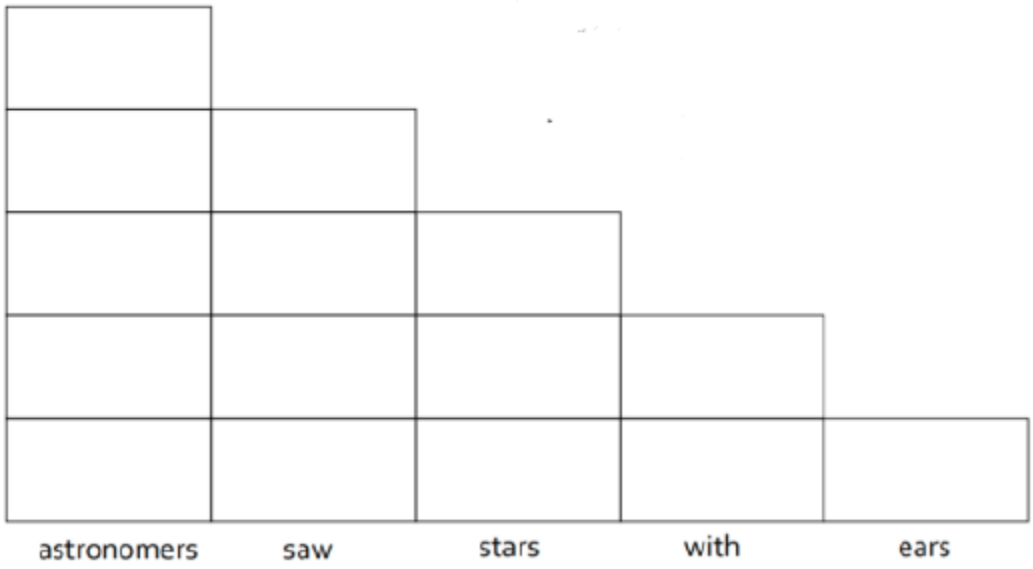
- S → NP VP (1.0)
- PP → P NP (1.0)
- VP → V NP (0.7)
- VP → VP PP (0.3)
- P → with (1.0)

- V → saw (1.0)
- NP → NP PP (0.4)
- NP → astronomers (0.1)
- NP → ears (0.18)
- NP → saw (0.04)
- NP → stars (0.18)
- NP → telescopes (0.1)

Sentence:

Astronomers saw stars with ears

CKY Table to Fill:



Note: You should answer this question in detail.

Problem 4

In this assignment, you will train a Logistic Regression model for sentiment analysis using the Stochastic Gradient Descent (SGD) algorithm to update parameters. The dataset contains multiple emotion classes, but we will focus on two specific classes:

- Class 0: Sadness
- Class 1: Joy

This means you are converting a multi-class classification task into a binary classification problem. Your model should distinguish between these two emotions and ignore other classes in the dataset (For more information check this link).

Please complete the Gradient Descent Classifier notebook.

Task Details

1. Dataset Preparation:

- Merge the training and validation sets into a single dataset for training.
- Retain only the samples corresponding to Class 0 (Sadness) and Class 1 (Joy), and discard samples belonging to other classes.

2. Model Training with SGD:

- Train a Logistic Regression model using the Stochastic Gradient Descent (SGD) algorithm. The training process should follow the below steps:

Stochastic Gradient Descent (SGD) Algorithm:

function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, x, y) **returns** θ
    # where: L is the loss function
    # f is a function parameterized by θ
    # x is the set of training inputs $x^{(1)}$, $x^{(2)}$, ..., $x^{(m)}$
    # y is the set of training outputs (labels) $y^{(1)}$, $y^{(2)}$, ..., $y^{(m)}$

$\theta \leftarrow 0$
repeat til done
  For each training tuple $(x^{(i)}, y^{(i)})$ (in random order)
    1. Optional (for reporting): # How are we doing on this tuple?
      Compute $\hat{y}^{(i)} = f(x^{(i)}; \theta)$ # What is our estimated output $\hat{y}$?
      Compute the loss $L(\hat{y}^{(i)}, y^{(i)})$ # How far off is $\hat{y}^{(i)}$ from the true output $y^{(i)}$?
    2. $g \leftarrow \nabla_\theta L(f(x^{(i)}; \theta), y^{(i)})$ # How should we move θ to maximize loss?
    3. $\theta \leftarrow \theta - \eta\, g$ # Go the other way instead
return θ

Assignment Criteria

a. Dataset Preparation: **(10 points)**
    - Correctly filter and merge the dataset to include only the specified classes.
    - Properly encode the labels for binary classification.

b. Model Implementation: **(15 points)**
    - Train the Logistic Regression model using SGD, adhering to the algorithm provided.
    - Ensure proper computation of the loss and gradient updates.

c. Code Clarity and Comments: **(10 points)**
    - Provide clear and concise code with meaningful comments explaining each step of the process.
    - Avoid importing external libraries—implement all logic from scratch.