Iran University of Science & Technology

School of Computer Engineering

Assignment #3

Natural language processing

BY:

DR. Behrouz Minaei, Fall 2024

Teaching Assistants:

Reza Alidoost

Due: 1403/09/05

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 HW3_StudentID.zip by the specified deadline.
2. A total of 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 4 days, and after 4 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

Suppose we have the following labeled sentences for training, where each sentence is labeled as either Positive (P) or Negative (N). **(25 points)**

State	Doc	Sentence	Class
Training	1	I love this movie	Positive
	2	This movie is fantastic	Positive
	3	I hate this movie	Negative
	4	This movie is boring	Negative
	5	I enjoy this movie	Positive
Test	6	I love this amazing movie	Positive
	7	This movie is amazing but boring and I hate it	Negative

a. Based on the training sentences, calculate the prior probabilities for each class (Positive and Negative). How do these priors influence the classification of a new sentence? **(5 points)**

b. Construct the vocabulary by listing all unique words in the training data. Then, calculate the likelihood of each word given the class (Positive or Negative) using the Naive Bayes formula. Use Laplace smoothing to handle any zero probabilities. **(7 points)**

c. In the Naive Bayes model, we assume each word appears independently of others given the class. Discuss whether this assumption seems realistic for the given sentences. How might it affect our results? **(3 points)**

d. Using the priors and likelihoods you calculated, find the posterior probability for each class (Positive and Negative) for each test sentence. Which class does the model assign to each test sentence? **(5 points)**

e. Explain the role of Laplace smoothing in this example. What would happen if we didn't use smoothing when encountering words like "amazing" in the test sentences? **(2 points)**

f. Imagine you had more Negative examples than Positive examples in the training data. How might this imbalance affect the priors, and how would the classifier tend to predict? **(3 points)**

Problem 2

You are developing a basic spelling correction tool that uses edit distance to suggest corrections for misspelled words. Given a dictionary of correct words, the tool should suggest the word with the smallest edit distance to the misspelled word. **(25 points)**

Dictionary = {"distance", "resistance", "insistence", "instance", "substance", "assistance", "persistence"}

A user types the word “distnace” (note the transposed letters "n" and "a"). Calculate the edit distance between the misspelled word and each dictionary word using the Levenshtein distance, and identify the word with the smallest edit distance.

a. What is the edit distance between “distnace” and each word in the dictionary? Show the step-by-step transformations for each calculation. **(10 points)**

b. If two or more words have the same edit distance, describe an additional criterion to choose the most likely correct word. Implement this criterion in your answer (e.g., based on word frequency or semantic similarity). **(5 points)**

c. Modify the edit distance calculation to assign different weights to different types of edits. For example:
   - Assign a higher cost for transpositions than for simple substitutions.
   - Assign a lower cost for inserting or deleting common letters like “e” or “s.”

   Recalculate the edit distances based on these new weights and determine the best match. **(10 points)**

Problem 3

We have a small dictionary with the following words: “there”, “their”, “they're”, “the”

You have a corpus of sentences from which you can compute the prior probabilities $P(word_{correct})$, based on their frequency of occurrence: **(20 points)**

- “there” appears 1000 times.
- “their” appears 400 times.
- “they're” appears 100 times.
- “the” appears 3000 times.

a. Calculate the prior probabilities $P(word_{correct})$ for each word in the dictionary based on their frequency in the training data. **(2 points)**

b. You need to define $P(word_{misspelled}|word_{correct})$ the probability that “their” is a misspelling of each dictionary word. Assume that this probability is based on the edit distance between the misspelled word and the candidate correct word, and use the following assumptions: (hint: You must define a probability function based on edit distance) **(8 points)**
   - A substitution, insertion, or deletion operation has a cost of 1.
   - A transposition operation has a cost of 0.5 (because letters in adjacent positions may be swapped, which is common in misspellings).

c. Using the Noisy Channel Model, calculate the most likely correct word for the misspelled word “thier.” Show all the calculations for $P(word_{correct}\ |"thier")$ for each candidate word. **(5 points)**

d. What would happen if the misspelled word was "there" (which could be confused with "their")? How would the model handle this ambiguity? What can be done to improve the model’s ability to handle such ambiguities in practice? **(5 points)**

Problem 4

In this part, we’re going to train a Naive Bayes Classifier for the task of sentiment analysis on the Hugging face emotion dataset (For more information check this link). Since this is a multi-class dataset and we want to train a binary classifier, we will use classes 0 and 1 (0:sadness, 1:joy). On the other side, You should merge the training set’s data with the validation set. Please complete the notebook provided in your assignment folder. (A report for your code is crucial) **(30 points)**

Your model should have accuracy above 90 percent on the test set.