Institute for Advanced Studies
in Basic Sciences
Gava Zang, Zanjan, Iran

Course: **Natural Language Processing**

Semester: **Spring 2024**

Department: **Computer Science and Information Technology**

Homework: **The BLEU Score**

Due Date: **May 20, 2024, by 23:59**

Submission: **Submit as a Source + Short Report File**

File Name: **FirstNameLastName_BS.zip**

Email for submission: **m.tareh@iasbs.ac.ir**

Project Assignment: Implementing the BLEU Score

Important Notes:

- Ensure your submission is entirely your own work. Refer to the university's policy on academic integrity (Cheating = 0!).
- You can copy and paste this text into a file, or just write answers on note paper, or whatever is easiest to work with for yourself (just keep studying!).
- Remember, GPT might help you 'pass', but only your own smarts will help you 'excel'!

Objective:

The objective of this project is to implement the BLEU score, a method for automatically evaluating machine-translated text. The implementation should calculate the BLEU score using 1-gram, bigram, trigram, and four-gram precision, along with a brevity penalty to account for overly short translations.

Background:

The BLEU score is a way to measure the quality of machine-generated translations using a reference translation. It calculates the precision of n-grams in the generated text that appear in the reference text and applies a penalty for translations that are too short. The BLEU score can range from 0 to 1, where higher scores indicate better translations.

Requirements:

- Write a Python function calculate_bleu(candidate, reference) where:
- candidate is the machine-translated text (a list of tokens).
- reference is the correct translation (also a list of tokens).
- Your function should compute the BLEU score considering 1-gram, 2-gram, 3-gram, and 4-gram precision.
- Include a brevity penalty in your BLEU score calculation to penalize overly short translations.

Instructions:

Tokenization: Begin by writing a function to tokenize both the candidate and the reference texts into words. This function should convert the texts into lower case and split them into lists of tokens (words).

N-gram Precision Calculation: Implement a function to calculate the n-gram precision. For each n-gram in the candidate text, check if it appears in the reference text and count these matches. The precision for each n-gram is then calculated as the ratio of matching n-grams to the total number of n-grams in the candidate text.

Brevity Penalty: Calculate the brevity penalty. If the length of the candidate translation is shorter than the reference, the penalty is applied as exp(1 - r/c), where r is the length of the reference and c is the length of the candidate.

BLEU Score Calculation: Combine the individual n-gram precisions using the geometric mean and multiply by the brevity penalty to obtain the final BLEU score.

Testing:

- Test your function using the following example:
- Candidate: "The quick brown fox jumps over the lazy dog."
- Reference: "A quick brown fox leaped over the lazy dog."
- Analyze how the BLEU score changes with different modifications to the candidate text.

Submission:

Submit a Python script (.py file) that defines the calculate_bleu function and includes comments explaining your code. Ensure that your script also contains test cases demonstrating the use of the function with the provided example.

Evaluation:

Your implementation will be evaluated based on the correctness of the BLEU score calculation and the clarity of your code. Comments and code readability will also be considered in your grade.