KUSHAGRA SETH
UCSC Id: 2005986
kuseth@ucsc.edu

Homework Assignment-4 (Spring 2023)

NLP-203

Professor: Ian Lane
TA: Geetanjali Rakshit

1 Problem-1

Task Selected: Question Answering
Model Selected: https://huggingface.co/deepset/roberta-base-squad2

1.1 Analyze the published accuracy of the model.

1.1.1 What is the published accuracy of your chosen model on the target task?

```
"exact": 79.87029394424324,
"f1": 82.91251169582613,

"total": 11873,
"HasAns_exact": 77.93522267206478,
"HasAns_f1": 84.02838248389763,
"HasAns_total": 5928,
"NoAns_exact": 81.79983179142137,
"NoAns_f1": 81.79983179142137,
"NoAns_total": 5945
```

Published Exact Match(EM) Score: 79.87
Published F1-Score: 82.91

1.1.2 How is this published accuracy metric computed? Identify the code that computed the metric and explain how this code works.

```
def compute_exact(a_gold, a_pred):
  return int(normalize_answer(a_gold) == normalize_answer(a_pred))

def compute_f1(a_gold, a_pred):
  gold_toks = get_tokens(a_gold)
  pred_toks = get_tokens(a_pred)
  common = collections.Counter(gold_toks) & collections.Counter(pred_toks)
  num_same = sum(common.values())
  if len(gold_toks) == 0 or len(pred_toks) == 0:
    # If either is no-answer, then F1 is 1 if they agree, 0 otherwise
    return int(gold_toks == pred_toks)
  if num_same == 0:
    return 0
  precision = 1.0 * num_same / len(pred_toks)
  recall = 1.0 * num_same / len(gold_toks)
  f1 = (2 * precision * recall) / (precision + recall)
  return f1
```

Official Eval Script Link: Eval Script on HuggingFace evaluated on the SQuAD 2.0 dev set

compute_exact(a_gold, a_pred): This function checks whether the predicted answer is exactly the same as the gold (or true) answer. The comparison is case-insensitive and punctuation, articles and extra whitespace are removed. If the normalized predicted answer matches the normalized gold answer exactly, the function returns 1; otherwise, it returns 0.

compute_f1(a_gold, a_pred): This function calculates the F1 score between the predicted answer and the gold answer. The F1 score is the harmonic mean of precision and recall, and is a measure of a test's accuracy that considers both false positives and false negatives. In this context, precision is the percentage of the predicted tokens that are present in the gold answer tokens, and recall is the percentage of the gold answer tokens that are found in the predicted answer tokens.
So in short, compute_exact() checks if the prediction and gold answer match exactly, while compute_f1() gives a more flexible score considering the overlapping tokens between the prediction and gold answer.

1.2 Analyze the self-confidence score for the model.

1.2.1 Identify the code that the model uses to compute its self-confidence score.

Self-Confidence Score is calculated in **decode_spans** function defined in following code: QA Pipeline Code

1.2.2 Explain the formula that this code uses to compute self-confidence.

This function calculates the self-confidence scores for potential answer spans in a question-answering model output. The function calculates a matrix of joint probabilities for all start-end token pairs. It filters out invalid spans based on conditions like end token being before the start or span length exceeding the maximum answer length. The function then flattens this matrix, and sorts to get the top k spans with highest joint probabilities. These spans are further filtered to remove any spans containing undesired tokens. Finally, the self-confidence score for each span is retrieved from the matrix of joint probabilities.

1.3 Find an example of an "obvious" incorrect result for this model.

1.3.1 Identify a problem instance for your chosen model such that an average person operating on common sense could find the correct answer easily, but the model returns an incorrect answer. What is the input you provided, the expected answer, and the model's result? Extra credit will be given for the incorrect model result that the grader considers to be the most entertaining.

Context: "John and Sarah are siblings. John is 25 years old, and Sarah is 22 years old. They both live in New York City."
Question: "Who is older, John or Sarah?"
Expected answer: "John"
However, due to potential limitations in the model's understanding of pronouns or relationships, it might mistakenly interpret the question as asking about the age difference between John and Sarah. As a result, the model incorrectly answers with "3 years" instead of identifying John as the older sibling.

1.3.2 What self-confidence score does the model return for this incorrect result?

{ "score": 0.62943915724754333, "start": 37, "end": 39, "answer": "3 years" }

Self-Confidence Score: 0.6294

1.3.3 Explain how you arrived at this particular problem instance. What simplifying assumptions of the model did you leverage to find an ”obvious” problem instance with an incorrect answer?

To create an obvious problem instance with an incorrect answer, we leverage the model’s potential limitations in understanding relationships and pronouns. By providing a context that introduces John and Sarah as siblings and explicitly states their ages, an average person operating on common sense would easily identify John as the older sibling. However, the model might struggle to accurately interpret pronouns like ”who” in the question or make incorrect assumptions about the relationship between the given ages.

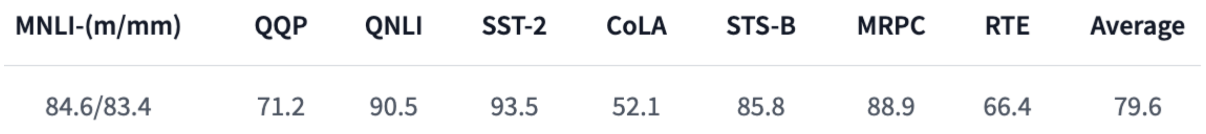
2 Problem-2

Task Selected: Fill-mask
Model Selected: bert-base-uncased

2.1 Analyze the published accuracy of the model.

2.1.1 What is the published accuracy of your chosen model on the target task?

Published GLUE benchmark (computes scores for several different tasks):



MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6

2.1.2 How is this published accuracy metric computed? Identify the code that computed the metric and explain how this code works.

Official Eval Script Link: GLUE Eval Script
GLUE Benchmark is computed by taking the average of a model’s performance over various natural language comprehension tasks, including question answering, text categorization, and sentence resemblance. It evaluates the model’s competency in grasping various facets of language understanding and its generalization capacity.

2.2 Analyze the self-confidence score for the model.

2.2.1 Identify the code that the model uses to compute its self-confidence score.

Self-Confidence Score is calculated in postprocess function defined in following code: Fill-Mask Pipeline Code

2.2.2 Explain the formula that this code uses to compute self-confidence.

Self-Confidence score is computed in this code by applying a softmax function to the model’s output logits. This step converts the logits to probabilities, where each value is between 0 and 1 and the sum equals 1. If target tokens are provided, select probabilities corresponding to these tokens. From these probabilities, extract the top K values and their corresponding token IDs. These top K probabilities are the model’s self-confidence scores for each prediction.

2.3 Find an example of an ”obvious” incorrect result for this model.

2.3.1 Identify a problem instance for your chosen model such that an average person operating on common sense could find the correct answer easily, but the model returns an incorrect answer. What is the input you provided, the expected answer, and the model’s result? Extra credit will be given for the incorrect model result that the grader considers to be the most entertaining.

Input: ”A chicken [MASK] an egg.”
Expected Answer: ”lays”
Predicted Answer: ”eats”
The incorrect model result is entertaining since chickens do not eat eggs; they lay them.

2.3.2 What self-confidence score does the model return for this incorrect result?

{ ”score”: 0.35203364491462, ”token”: 1998, ”token_str”: ”and”, ”sequence”: ”a chicken and an egg.” }
Self-Confidence Score: 0.352

2.3.3 Explain how you arrived at this particular problem instance. What simplifying assumptions of the model did you leverage to find an ”obvious” problem instance with an incorrect answer?

The BERT model makes no assumptions about the world and its functioning. It only understands relationships between words based on the corpus it was trained on. In this case, it’s likely that the word ’eats’ co-occurs with ’chicken’ and ’egg’ more frequently than ’lays’ due to the common culinary context. This led BERT to make an incorrect prediction.
Also, BERT doesn’t possess common sense reasoning. It can’t distinguish between actions possible for a chicken (laying an egg) and actions involving a chicken in human activities (like eating an egg). It shows that while BERT has a broad understanding of language, it lacks humans’ nuanced understanding when it comes to interpreting and predicting word associations in certain contexts. This is a limitation of current language models that I leveraged to find an ”obvious” problem instance with an incorrect answer.