Introduction to BERTScore

T Ravindra

July 30, 2024

Introduction to BERTScore

- **What is BERTScore?**
  BERTScore calculates a score that measures how similar each word in your text is to each word in a reference text. Unlike older methods, it doesn't just look for exact word matches.
- **Contextual Embeddings and Semantic Similarity:**
  BERTScore uses contextual embeddings that consider the context of each word, enabling a deeper understanding of text meaning. This approach marks a significant improvement over traditional methods like n-gram overlap, edit distance, and basic embedding matching, which mainly focus on surface-level similarities.
- **Why Use BERTScore?**
  It aligns more closely with human judgment compared to previous methods, making it a more reliable tool for evaluating text generation.

Prior Work in Text Evaluation

- **N-gram Overlap:**
  This method uses string matching, and cannot capture relationships between words that are far apart due to their limited window size.
- **Edit Distance:**
  This metric measures how many edits (like adding, removing, or substituting letters) are needed to turn the candidate text into the reference text. It focuses solely on surface-level changes, not on the meaning of the text.
- **Embedding Matching:**
  This approach uses word embeddings to evaluate lexical and structural similarities but does not account for the context of words. Relying on external tools for generating embeddings also complicates reproducibility.
- **Learned Functions:**
  These are regression models that combine various metrics but require extensive labeled data, making them expensive to train and hard to generalize across different tasks.

BERTScore Computation Method

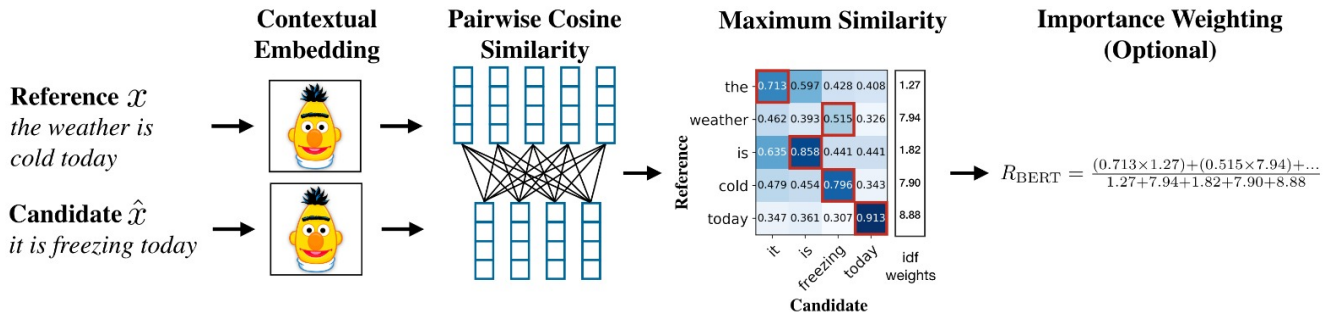

Figure: Illustration of BERTScore computation

- Given a reference sentence $x = \langle x_1, x_2, \ldots, x_k \rangle$ and a candidate sentence $\hat{x} = \langle \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_l \rangle$, BERTScore uses contextual embeddings from various BERT models to represent the tokens, and compute matching using cosine similarity, optionally weighted with inverse document frequency scores.

BERTScore Metrics

Greedy Matching

Greedy matching is used to maximize the similarity score for each token. The recall, precision, and F1 scores are given by:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{i \in x} \max_{j \in \hat{x}} (x_i^\top x_j),$$

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{j \in \hat{x}} \max_{i \in x} (x_i^\top x_j),$$

$$F_{\text{BERT}} = \frac{2 \cdot P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}.$$

Importance Weighting

Importance weighting is used to give more importance to tokens that are rare in the corpus. The importance weight is calculated by:

$$\text{idf}(w) = -\log \frac{1}{M} \sum_{i=1}^{M} [w \in x(i)],$$

where M is the number of reference sentences in the corpus and $x(i)$ is the i-th sentence in the corpus.

Baseline Rescaling

The scores are rescaled to be between 0 and 1 using the following formula:

$$\hat{R}_{\text{BERT}} = \frac{R_{\text{BERT}} - b}{1 - b}$$

where b is a baseline score determined from validation data to normalize scores across different evaluations.

Main Strengths of the BERTScore Paper

Semantic Understanding

BERTScore leverages BERT embeddings to capture the contextual and semantic similarity between texts, going beyond surface-level matches like traditional metrics (BLEU, ROUGE, METEOR).

Robustness to Paraphrasing

BERTScore is robust to paraphrasing and variations in wording, which are common in natural language.

Major Weaknesses and Suggested Improvements

Major Weaknesses

- **Computational Complexity:** The computation of BERTScore is more resource-intensive compared to traditional metrics.
- **Dependence on Pre-trained Models:** BERTScore relies heavily on pre-trained BERT models, which may not be optimal for all languages or domains.

Suggested Improvements

Fine-tune the BERT model on domain-specific data to improve its performance in specialized contexts.