BERTSCORE: EVALUATING TEXT GENERATION WITH BERT

By – Kriti Banka
BML Munjal University, Gurgaon

Summary

Traditional methods of evaluating natural language generation rely heavily on surface-form similarity and often fails to accurately capture semantic equivalence. For example, BLEU, the most commonly machine translation metric, simply counts *n*-gram overlap between the candidate and reference and METEOR is again an evaluation of machine translation output. The metric is based on the harmonic mean of unigram precision and recall, with the recall weighted higher than precision. To understand BLEU score and its term lets, understand the procedure to calculate BLEU Score.

BERTScore is a significant metric that has emerged as an alternative to traditional evaluation metrices in the field of natural language processing. It's a language generation evaluation metric based on pre trained BERT (Bidirectional Encoder Representations from Transformers) contextual embeddings. BERTScore computes the similarity of two sentences as a sum of cosine similarities between their tokens' embeddings. In contrast to BLEU, BERTScore is not restricted to maximum n-gram length, but instead relies on contextualized embeddings that are able to capture dependencies of potentially unbounded length. METEOR requires external resources, only five languages are supported with the full feature set, and eleven are partially supported. Similar to METEOR, BERTSCORE allows relaxed matches, but relies on BERT embeddings that are trained on large amounts of raw text and are currently available for 104 languages. BERTScore also supports importance weighting, which we estimate with simple corpus statistics.

BERTScore Methodology

Given a reference sentence x = (x_1, . . ., x_k) and a candidate sentence $\hat{x}$ = (x_1, . . ., x_l), BERTScore use contextual embeddings to represent the tokens, and compute matching using cosine similarity, optionally weighted with inverse document frequency scores.

Token Representation

Contextual embeddings such as BERT and ELMo, use to represent tokens in input sentences x and $\hat{x}$ highlighting their ability to generate different vector representations for the same word in different sentences depending on the surrounding words, which form the context of the target word. The generated embeddings are most commonly trained using various language modeling objectives, such as masked word prediction. The main model we use is BERT, which tokenizes the input text into a sequence of word pieces. The representation for each word piece is computed with a Transformer encoder by repeatedly applying self-attention and nonlinear transformations

in an alternating fashion.

Cosine Similarity

The core of BERTScore approach is to compute the cosine similarity between each pair of tokens from the reference and candidate sentences, capturing nuanced semantic relationships.

To complete score matches each token in x to a token in $\hat{x}$ to compute recall, and each token in $\hat{x}$ to a token in x to compute precision. We use greedy matching to maximize the matching similarity score, where each token is matched to the most similar token in the other sentence

Importance Weighting

Tokens can be weighted differently based on their importance, which can be estimated using inverse document frequency (IDF) scores computed from the test corpus.

There is a limited numerical range due to the learned geometry of contextual embeddings. To make the scores more interpretable, baseline rescaling method is used, where scores are rescaled to the use of an empirically derived cut lower bound *b,* calculated from BERTScore on random sentence pairs from the Common Crawl dataset. This rescaling makes the scores values reflect a normalized quantity and therefore improves the interpretability of BERTScore itself, without losing the ability to stratify text generation systems.

Experiment Setup

The experiment setup focuses on evaluating the BERTSCORE metric across multiple tasks, specifically machine translation and image captioning. A variety of pre-trained contextual embedding models like BERT, RoBERTa, XLNet, and XLM are applied to several datasets.

Contextual Embedding Models: 12 pre-trained models were tested, including BERT, RoBERTa, XLNet, and XLM. Few models used were based on the language for say, BERTchinese for Chinese, RoBERTa for English. The WMT16 dataset was used for layer selection.

Machine Translation: Main evaluation corpus is the WMT18 metric evaluation dataset, which contains predictions of 149 translation systems across 14 language pairs, gold references, and two types of human judgment scores. WMT18 includes translations from English to Czech, German, Estonian, Finnish, Russian, and Turkish, and from the same set of languages to English. The evaluation matrices were Pearson correlation Kendall rank correlation, and statistical tests. BERTSCORE was compared against other metrices like BLEU, with results showing high correlation with human judgement with FBERT giving highest correlation.

Image Captioning: used the human judgments of twelve submission entries from the COCO 2015 Captioning Challenge. BERTScore was compared with task-agnostic and task-specific metrices. Pearson correlation with system-level metrices showed BERTScore's effectiveness.

Key Results

1. Machine Translation:
    - *System-Level Correlation:* BERTSCORE is consistently a top performer that shows strong correlation with human judgments, often outperforming other metrics like BLEU, especially in cases involving hybrid systems.
    - *Segment-Level Correlation:* BERTSCORE significantly outperforms traditional metrics like SENTBLEU, making it more reliable for analyzing specific examples.
    - *Importance Weighting:* IDF weighting sometimes provides small benefits but is not universally helpful. The F1 score (FBERT) is recommended as a reliable metric across different settings.
2. Image Captioning:
    - *Performance:* BERTSCORE outperforms all task-agnostic baselines by large margins showing strong correlations with human judgments in challenging evaluation scenarios like the COCO Captioning Challenge.
    - *Statistics*: IDF importance weighting showed significant benefit for this task. IDF weighting achieves a Pearson correlation of 0.917 significantly higher than other metrices like BLEU (-0.019) and ROUGHE-L (0.090).
3. Speed:
    - Despite the use of a large pre-trained model, computing BERTSCORE is relatively fast. BERTScore is able to process 192.5 candidate-reference pairs/second using a GTX-1080Ti GPU. The complete WMT18 en-de test set, which includes 2,998 sentences, takes 15.6sec to process, compared to 5.4sec with SacreBLEU (Post, 2018), a common BLEU implementation.
4. Robustness Analysis (Adversarial Paraphrase Classification):
    - *Datasets:* Tested on QQP and PAWSQQP datasets, BERTSCORE demonstrates greater robustness to adversarial examples compared to traditional metrics and even supervised classifiers like BERT fine-tuned on QQP.
    - *Statistics:* On the PAWSQQP dataset, BERTSCORE achieves an AUC of 0.693 with IDF weighting, showing a relatively small performance drop compared to other metrics, which struggle with these harder adversarial examples.

Three Major Strengths of the paper

1. **Semantic Evaluation Over Surface Form**:
   Unlike traditional metrices like BLEU the focus on n-gram overlap, BERTScore evaluates semantic using contextual embeddings, making it more robust to meaning-preserving lexical and syntactic variations. Also, correlation between BERTScores with human evaluations is better across different task.

2. **Multiple Languages Support**:
   BERTScores leverages BERT embeddings, which are available for 104 languages, making it a versatile tool for evaluating text generation across different linguistic contexts.

3. **Flexibility and Simplicity**:
   BERTScore can be easily implemented and does not require external resources like syntactic parsers or manually curated lexicons, unlike some other metrics such as METEOR.

Three Major Weaknesses of the paper

1. **Dependence on Pre-trained Models:**
   BERTScores relies on Pre-trained BERT models, which may not generalize well to domains that the model was not trained on. This limits its applicability to tasks or languages that are not well represented in the BERT training data.

2. **Dependence on Similarity Threshold:**
   BERTSCORE uses cosine similarity between embeddings, but the choice of similarity thresholds or the handling of near-similar tokens can significantly impact the results. The paper does not fully explore how these thresholds affect performance across different scenarios.

3. **Computational Complexity**:
   The use of contextual embeddings and pairwise token comparisons increases the computational cost compared to simpler metrics like BLEU. This may not be feasible for real-time applications or for evaluating large datasets.

Three improvements to the paper

1. **Clarify Methodological Assumptions:**

   There can be a clearer explanation of the assumptions underlying BERTScore, about varying sentences structures and how it handles different languages. Providing more detailed examples of edge cases or challenging scenarios would strengthen the readers' understanding of the metric's robustness.

2. **Detail on Implementation and Reproducibility**:

   Providing in-depth details on the implementation, including any preprocessing steps, hyperparameters, and potential challenges in reproducing the results, would be beneficial.

3. **Address Limitations and Future Work**:

   There can be thorough discussion of the limitations of BERTSCORE, especially in cases where it may not perform well. Additionally, outlining potential future work, such as improvements to the metric or new areas of application, would provide a clear direction for ongoing research.