**BERTSCORE: EVALUATING TEXT GENERATION WITH BERT**

**SUMMARY**

**ABSTRACT:**

BERTSCORE is an **automatic evaluation metric** for text generation. It computes a similarity score for each token in the candidate sentence with each token in the reference sentence. Instead of exact matches, token similarity is computed using **contextual embeddings**. Outputs of 363 machine translation and image captioning systems were evaluated and it is obsereved that **BERTSCORE correlates better with human judgments and provides stronger model selection performance than existing metrics.**

**INTRODUCTION:**

One significant advantage it offers over traditional n-gram-based metrics like BLEU is its **ability to handle paraphrases better**. N-gram methods often struggle with paraphrases, leading to underestimation of performance when semantically correct phrases are penalized due to differences from the reference's surface form. BERTSCORE addresses this by utilizing **contextualized token embeddings**, which are known to be **effective for detecting paraphrases.**

Overall, BERTSCORE seems like a valuable addition to the toolkit for evaluating natural language generation tasks like **machine translation and caption generation**, offering a more **nuanced and accurate assessment of semantic similarity** compared to traditional methods.

**PROBLEM STATMENT:**

Natural language text generation is commonly evaluated using annotated reference sentences. Given a reference sentence x tokenized to k tokens <x1, . . . , xk> and a candidate ˆx tokenized to l tokens <ˆx1, . . . , ˆxl>, a generation evaluation metric is a function f (x, ˆx) ∈ R. Better metrics have a higher correlation with human judgments. Existing metrics can be broadly categorized into using n-gram matching, edit distance etc. Traditonal method vs BERTSCORE follows:

1. **SIMILARITY SCORE**: The vector representations offer a flexible similarity measure compared to strict string or heuristic matching methods. We compute the cosine similarity between each reference token and candidate token, simplifying the calculation with pre-normalized vectors to an inner product. Although this measure evaluates tokens individually, contextual embeddings incorporate information from the entire sentence, enhancing the overall similarity assessment.
2. **BERT SCORE**: BERTSCORE calculates the complete score by matching each token in the reference sentence with a token in the candidate sentence to compute recall, and vice versa for precision. We use greedy matching to maximize the similarity score, ensuring each token is matched with its most similar counterpart in the other sentence.

Given a reference sentence x = 〈x1, . . . , xk〉 and a candidate sentence ˆx = 〈ˆx1, . . . , ˆxl〉, we use contextual embeddings to represent the tokens, and compute matching using cosine similarity, optionally weighted with inverse document frequency scores.

These models are typically trained with language modeling objectives such as masked word prediction. We experiment with various models and their tokenizers, like BERT, which tokenizes input text into word pieces and computes representations using a Transformer encoder. BERT embeddings have shown benefits across different NLP tasks due to their ability to capture contextual information effectively.

**EXPERIMENTAL SET UP:**

1. **MACHINE TRANSLATION**: The main evaluation dataset used is the WMT18 metric evaluation dataset, containing predictions from 149 translation systems in 14 language pairs, human evaluations, and gold references. The evaluation includes **translations between English and various languages** such as Czech, German, Estonian, Finnish, Russian, and Turkish. **Evaluation metrics used include absolute Pearson correlation and Kendall's rank correlation** to assess the quality of the metric system.
2. **IMAGE CAPTIONING**: Traditional metrics like BLEU don't correlate well with human judgement. **BERTSCORE shows promise for better evaluation**, while idf weighting suggests focusing on content words.
3. **SPEED:** Despite the use of a large pre-trained model**, computing BERTSCORE is relatively fast**.Given the sizes of commonly used test and validation sets, the increase in processing time is relatively marginal, and BERTSCORE is a good fit for using during validation and testing, especially when compared to the time costs of other development stages.

**RESULTS:**

BERTSCORE performs well in **machine translation** evaluations, particularly **surpassing other metrics like BLEU and RUSE**. While RUSE shows competitive results, it relies on specific training data and may not be applicable in all scenarios. Additionally, using idf weighting occasionally improves performance but is not universally beneficial, suggesting its usefulness depends on text domain and available data. F1 score is recommended as a reliable measure across different settings.

In **image captioning**, BERTSCORE excels, showing significant improvements over task-agnostic baselines, while idf weighting proves beneficial, **indicating the importance of content words**. Despite being based on a large model, BERTSCORE computation remains relatively fast, making it suitable for validation and testing stages in development.

**a.What are the three major strengths of the paper?**

1. BERTSCORE uses the contextual embeddings from BERT to compare sentences. This helps it understand the meaning and connections between words better than simpler methods like n-grams.

2. Compared to other metrics, BERTSCORE aligns more closely with human judgments in tasks like machine translation and image captioning.

3. BERTSCORE works with 104 different languages because BERT was trained on a lot of diverse text. It's also flexible, allowing for adjustments like importance weighting and rescaling to make evaluations more accurate and understandable.

**b. What are the three major weaknesses of the paper?**

1. BERTSCORE is better at recognizing paraphrases than simpler methods, but it can still struggle with tricky paraphrases.
2. BERTSCORE tries to understand how words relate to each other, even when they're far apart. However, it might not always get the meaning right when the word order changes a lot.
3. BERTSCORE performs well against tricky examples in tests, but it's not completely immune to all kinds of tricky situations or adversarial attacks.

**c. Suggest three improvements to the paper.**

1. Test BERTSCORE with more languages, including those with fewer resources, to make it work better with a wider variety of texts.
2. Look into adding BERTSCORE into the training process of models so that they learn to optimize for evaluation metrics directly.
3. See how using specialized BERT models, like SciBERT for science texts, could make evaluations more accurate in specific areas..