BERTSCORE:
EVALUATING TEXT GENERATION WITH BERT

Tianyi Zhang, Varsha Kishore, ,FelixWu, Kilian Q. Weinberger,and Yoav Artzi

Background

- Commonly used metrics in language generation tasks
  - Evaluates based on n-gram overlap, eg. BLEU.
  - Fail to capture long dependencies.
  - Fails to penalize ordering changes, eg. "*Clouds caused rain", "Rain caused clouds".*

Proposed Methodology

Given a reference sentence, $x =< x_1, ..., x_k >$

and a candidate sentence, $\hat{x} =< \hat{x_1}, ..., \hat{x_l} >$

BERTScore computes similarity of two sentences as *sum of cosine similarities* between their *contextualized* token embeddings.

Strengths

- Different words conveying the same thing, would be judged based on their meanings, rather than exact match.
- Not restricted to capturing dependencies upto n-gram length.
- Token level matching allows weighing tokens according to their importance.

Weaknesses

- One particular configuration of BERTScore does not work bet for all tasks.
- BERTScore evaluates based on the order of words. It might not be very suitable for tasks where order of words does not matter.
- BERTScore depends on tokens. It might be misleading if the sentences are not tokenized properly.

Possible Improvements

- Embeddings could be generated using models other than BERT

- A parameter could be introduced which lets the user control whether BERTScore should consider the ordering of words or not.

- Can come up with a customized tokenization technique especially for tasks to be evaluated using BERTScore

Thank You!