A Appendices

A.1 Experimental Details

Setup All experiments were run using the TextAttack framework in Jupyter notebooks running in Google Colab using Tesla K80 GPUs. [4].

Models The attacked models are pretrained models provided by TextAttack (Morris et al., 2020b). BERTScore and the Universal Sentence Encoder are also loaded through TextAttack. The pretrained models are available on the HuggingFace model hub under the following names:

- SNLI Dataset
  - textattack/bert-base-uncased-snli
  - textattack/albert-base-v2-snli
  - textattack/distilbert-base-cased-snli
- SST-2 Dataset
  - textattack/bert-base-uncased-SST-2
  - textattack/albert-base-v2-SST-2
  - textattack/distilbert-base-cased-SST-2
- Rotten Tomatoes Dataset
  - textattack/bert-base-uncased-rotten-tomatoes
  - textattack/albert-base-v2-rotten-tomatoes
  - textattack/distilbert-base-uncased-rotten-tomatoes

A.2 Constraints tested on paraphrase datasets

Before running adversarial attacks on USE and BERTScore, we compared their effectiveness on common paraphrase identification tasks.

USE and BERTScore each assign a semantic similarity score to each (original text, perturbed text) pair. A hard threshold determines whether a given score indicates a valid adversarial example. Above this threshold, the perturbed text is assumed to have preserved the semantics of the original input; below it, semantics is not preserved, and the perturbation is invalid. Li et al. (2018) defines validity as a cosine similarity of 0.8 or higher, as measured by USE. Jin et al. (2019) and Garg and Ramakrishnan (2020) choose a lower USE threshold of 0.5.

Current state-of-the-art attacks in NLP generate perturbations one word at a time: generally by swapping out a word with neighbors in the embedding space (Alzantot et al., 2018) or with synonyms provided by a thesaurus (Ren et al., 2019). Consequently, their adversarial perturbations share the lexical structure of the original inputs, with some words swapped out for synonyms. This implies that BERTScore would be a better fit for ensuring semantic preservation during these adversarial attacks, and less susceptible to second-order adversarial examples.

Our initial question was how USE and BERTScore compare on common datasets for paraphrase identification. When used as constraints on adversarial attacks, constraints that can more correctly distinguish paraphrases from non-paraphrases should be less vulnerable to second-order adversarial examples.

In the following subsections, we compare USE and BERTScore on two paraphrase datasets, QQP and PAWS, and then on Adversarial SNLI, on a custom dataset designed to resemble the format of NLP adversarial examples on the SNLI entailment dataset.

A.2.1 Performance on paraphrase identification

We evaluate USE and BERTScore on two common paraphrase datasets:

- The **QQP (Quora Question Pairs)** dataset, which contains 400k real-world pairs of paraphrases and non-paraphrases collected during Quora question disambiguation.

- The **PAWS (Paraphrase Adversaries from Word Scrambling)** dataset, which contains 100k paraphrases and non-paraphrases. These examples originally come from the QQP paraphrases; non-paraphrases have been adversarially edited to change semantics while retaining high lexical overlap from the source. (Yang et al., 2019)

We sampled 1000 examples from the QQP and PAWS test sets. All datasets are loaded using the `nlp` package from HuggingFace[5]. The TextAttack

---

[4] Google Colab is a great resource, providing free, easy access to high-powered GPUs, but its timeout constraints can be frustrating and unpredictable. By the end of the project, this author shelled out $9.99 for the high-octane *Google Colab Pro*.

[5] See https://github.com/huggingface/nlp.

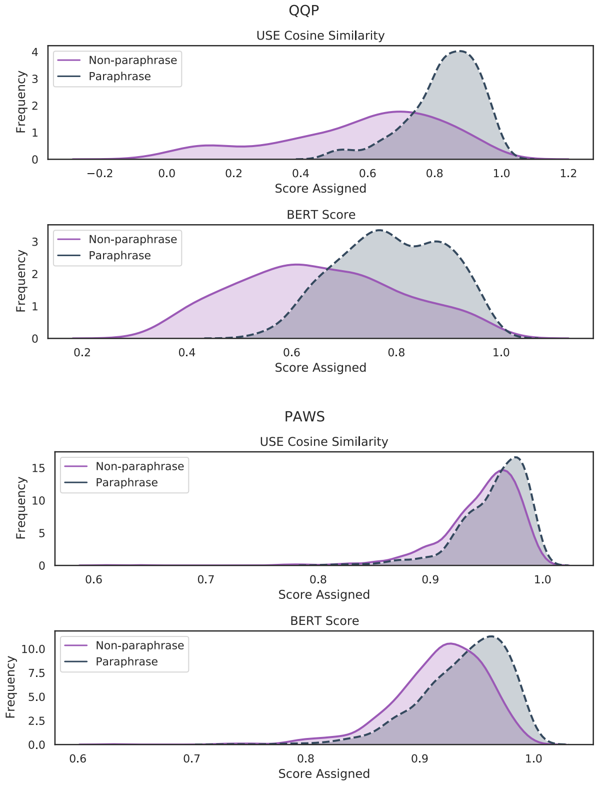

Figure 6: Distribution of scores assigned by BERTScore and the Universal Sentence Encoder (USE) on the QQP and PAWS datasets.

library (Morris et al., 2020b) is used to load pretrained USE and BERTScore models and to run augmentation and adversarial attack experiments.

Figure 6 shows the distributions of scores from each model (USE, BERTScore) on each dataset (QQP, PAWS). Both models exhibit some ability to distinguish paraphrases and non-paraphrases on QQP, but produce very similar scores for paraphrases and non-paraphrases on PAWS (with the non-paraphrases having slightly lower scores).

We then used these scores to plot ROC curves for each dataset; these are shown in Figure 7. Table **??** shows AUC for each model and dataset. Surprisingly, USE (AUC 0.827) slightly outperforms BERTScore (AUC 0.764) on QQP; however, BERTScore (AUC 0.662) outperforms USE (AUC 0.608) on the PAWS dataset. This corroborates findings from Zhang et al. (2019) that BERTScore is superior to sentence encoding methods on datasets with high lexical overlap.

A.2.2 Performance on Adversarial SNLI

BERTScore exhibited higher performance than USE on PAWS, a dataset of adversarial crafted

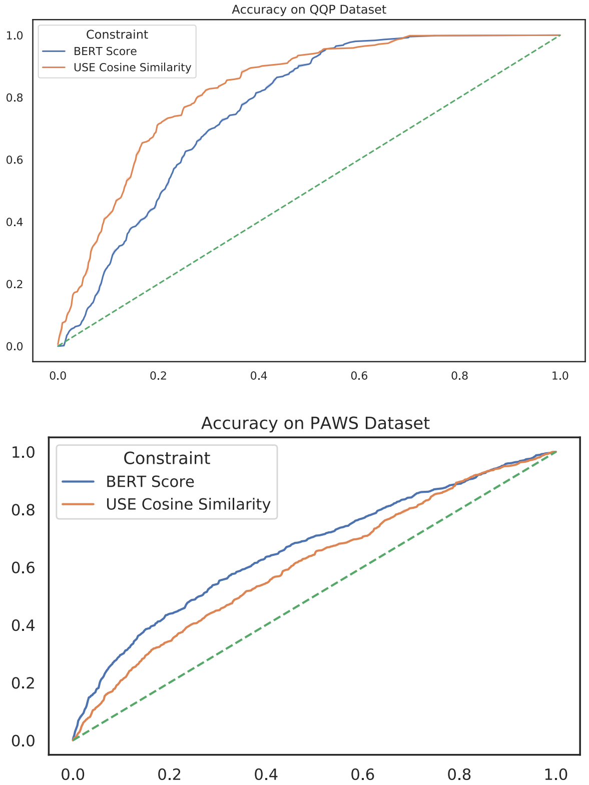

Figure 7: ROC Curves for BERTScore and the Universal Sentence Encoder (USE) on the QQP and PAWS datasets. USE outperforms BERTScore on QQP, but BERTScore is better at PAWS.

Dataset	USE	BERTScore
QQP	0.827	0.764
PAWS	0.608	0.662
Adversarial SNLI	0.635	0.710

Table 3: AUC Scores for BERTScore and the Universal Sentence Encoder on QQP, PAWS, and our Adversarial SNLI dataset. BERTScore shows an advantage PAWS and Adversarial SNLI, indicating that it is a more robust choice for constraining semantics during NLP adversarial example generation.

paraphrases. However, USE outperformed on QQP, a more traditional paraphrase task. To shed light on which method might perform better in an NLP attack setting, we generate a dataset that resembles potential perturbations during an NLP attack.

We set out to compare the two constraints in a scenario more similar to a typical NLP adversarial attack. To do this, we crafted a dataset of perturbations that might appear during the course of an adversarial attack.

We crafted our dataset of adversarial perturbations starting with examples from the SNLI dataset. We chose SNLI because it is commonly used for testing NLP adversarial attack systems (Zhang et al., 2020b), and because second-order adversarial examples are particularly dangerous in the case of entailment, where a slight change in meaning can cause a shift in ground-truth output. However, this process could be emulated to test out constraint options before running an adversarial attack on any NLP dataset.

We sampled 1,000 (premise, hypothesis) from the SNLI dataset and discarded each premise. For each hypothesis, we created ten adversarial examples: one by substituting synonyms, and one by substituting antonyms, and by substituting each of $(10\%, 20\%, 30\%, 40\%, 50\%)$ of the original words. This produced a dataset with 10,000 examples. We sourced synonyms and antonyms from WordNet (Miller, 1995).

BERTScore achieved a higher AUC on the two adversarial datasets, PAWS and Adversarial SNLI. This is a surprising result since BERTScore turned out to be so much less effective than USE as a constraint on adversarial examples (see Section 5). We hypothesize that BERTScore is better at measuring semantic changes of 1-2 words, while USE is superior as the perturbation size grows beyond 2 words.

We can also see how across datasets, BERTScore assigns scores that are generally lower; a threshold of $\epsilon = 0.8$ on USE cosine similarity may correspond to a lower threshold, for example, $\epsilon = 0.5$.