Bertology 101: Studying the Stability of the BERT Semantic Space

Testing BERT on a similarity benchmark

In this exercise, you will be comparing BERT embeddings drawn from two different corpora on a word-type similarity benchmark.

1. Download data

- **Python code for loading BERT:** Download the original repository using `git clone https://github.com/google-research/bert.git`. *It is highly recommended to use a dedicated virtual environment, eg. using `python3 -m venv bert-venv`.*
- **Download the model itself:** All information pertaining to this step should be on the github you installed in the previous step.
- **Retrieving a similarity benchmark:** Download the MEN dataset from https://staff.fnwi.uva.nl/e.bruni/MEN, and retrieve the file `MEN_dataset_natural_form_full`. It contains space-separated triples, composed of two words and a similarity rating.
- **Retrieve sentence corpora:** BERT embeddings are computed "on the fly", so there is no file containing the exact embeddings (unlike word2vec for instance). As a result, you need sentences corpora to compute embeddings from: retrieve the two pre-parsed corpora from the lecture's github (derived from Wikipedia and OpenSubtitles).

2. Retrieve word type representation

- **Retrieve word token embeddings:** Use the script `extract_features.py` from Google's BERT github. **Carefully read the README file for this github**. You only need the output from the last layer (use `--layers=-1`. This script should produce a JSON output that contains all the required information.
- **Compute word type representations:** Parse the output file from the previous step to retrieve individual tokens paired with their embeddings. You can then compute the average embedding for a given word type to retrieve a word type representation.

Tip: *do not retrieve all embeddings before computing the average for a given word type: instead, compute the sum of token embeddings as you parse them, and divide by the number of tokens for that word type.*

3. Compare embeddings from the two corpora

- **Using the MEN benchmark:**
  - Make sure you computed word-token representations: the script `extract_features.py` produces outputs associated to *word-pieces*.
  - For each triple from the MEN dataset, retrieve the word-type representations of the two paired words (as derived from the first of the two corpora)
  - Compute their cosine. This should result in a series of similarity measurements.
  - Compute the Spearman correlation of cosine measurements and human similarity ratings from MEN (i.e., compare the cosine with the third item from the triple).
  - Repeat the process, this time using embeddings from the second of the two copora.
- **Directly compare the embeddings of the two corpora:**
  - This time, you can directly compare embeddings of word-pieces.
  - Compute the average type representations for each word-piece embedding in the first corpus; repeat on the second corpus.
  - Compare directly the vectors component-wise using a related sample t-test over the two sets of vectors (use the scipy function scipy.stats.ttest_rel).
  - Probe the structure of the two vector spaces for the two corpora: a) select a random sub-sample of word-pieces, b) compute their pairwise distances (Euclidean or cosine) in both corpora, c) compute a t-test using these two related series of measurements.
- **What do you conclude from these experiments?**
  - How do you interpret them?
  - Are there unclear/uncertain points remaining?
  - How would you try to clarify them?