Sentence-BERT

Link: https://arxiv.org/pdf/1908.10084.pdf

BERT with **similarity comparison, clustering, and information retrieval.**

Better than InferSent, and Universal Sentence Encoder.

SBERT uses **siamese and triplet network** for training, data set is **SNLI and Multi-Genre NLI.**

- In Siamese networks, input image find encodings, use the same network (no change in weights or biases) and predict encodings of image of different person
- Train the network using an anchor image, compare to positive and negative results, dissimilarity should be low and high respectively.

$$\mathcal{L} = max(d(a,p) - d(a,n) + margin, 0)$$

- margin is added to the loss, how far away the dissimilarities must be.

Why SBERT?

Sentence Pair Regression, two sentences concatenated using seperator and a label is predicted. BERT cannot do independent sentence embeddings. Researchers passed single sentences through BERT and then

- derive a fixed sized vector by either averaging the outputs (average word embeddings) or
- by using the output of the special CLS token

Sentence Embeddings can be done through,

- Skip-Thought, unsupervised EN-DE to predict surrounding sentences,
- InferSent, using **Stanford NLI** dataset, to train siamese BiLSTM network with max-pooling over the output,
- Universal Sentence Encoder, trf and augments unsupervised learning with training on **SNLI**,

Poly-encoders, compute score between m vectors, pre computed candidate encodings using. attention, but score function is not symmetric and the computational overhead is too large for use-cases.

SBERT adds a **pooling operation** to the output of BERT / RoBERTa to derive a fixed sized sen- tence embedding,

- output of the CLS-token,
- the mean of all output vectors,
- computing a max-over-time of the output vectors

SBERT has three objective functions,

- **Classification Objective Function,** concatenate embeddings of two sentences, and element-wise difference, **(u,v,|u-v|), (relevant for training the soft- max classifier, during predictions, used is cosine similarity with embeddings).** and multiply with **trainable weights**, and optimize **cross-entropy loss**,
- **Regression Objective Function.** The cosine-similarity between the two sentence embeddings u and v is computed. We use **mean-squared-error** loss as the objective function.
- **Triplet Objective Function.** Given an anchor a, p, n, triplet loss tunes the network such that
  the dist(a and p) << dist(a and n), margin is 1.

Regression functions work pair-wise, arent scalable when the combination of sentences are large in number. SBERT uses cosine similarity between sentences,

Evaluation Strategy,

- Unsupervised STS, datasets used provide 0 and 5 on the semantic relatedness of sentence pairs. Authors use Spearman’s rank correlation between the cosine-similarity of the sentence embeddings and the gold labels.
- Supervised STS, regression objective function is used to train, at prediction time, we compute the cosine-similarity
- Argument Facet Similarity, similar sentences but lexically different, many models perform bad,
    - 10 cross fold validation, not clear how well approaches generalize to different top- ics.

    - cross topic setup, Two topics serve for training and the approach is evaluated on the left-out topic, average of all three topics, results show this is better.
- Wikipedia, distinct sections focusing on certain aspects, anchor section, p and n section.

In evaluation, sentence embeddings are used as feature for logistic regression. STS tasks use cosine-similarity to estimate the similarities, SentEval, fits a logistic regression classifier to the sentence embeddings meaning certain dimensions can have higher impact.