**Motivation**

- How do we search for similar, potentially lengthy, phrases in a database of N=100b+ phrases?
- We used transfer learning to fine tune the BERT model [4] using a fully connected single layer trained on labeled paraphrase pairs from to get fixed length representations of sentences.
- These fixed length vectors are then used in approximate nearest neighbor searches based on cosine similarity.

**Datasets**

- The validation dataset is 89 pairs of sentences with similarity 0.
- Another dataset was constructed pairing the first sentence with itself.
- The dataset with all pairs similar was separated into 4 clusters. Clusters obtained with bert base embeddings helped to form clusters. Clusters obtained with bert base embeddings:
  - Cluster 1: similar sentences from the original validation dataset with pair-wise similarity less than 0.5.
  - Cluster 2: sentences from the original validation dataset with pair-wise similarity greater than 0.5.
  - Cluster 3: sentences from the original validation dataset with pair-wise similarity less than 0.5.
  - Cluster 4: sentences from the original validation dataset with pair-wise similarity greater than 0.5.

**Models**

- Baseline: Averaged word2vec
- Bert base uncased (all lower case) [3]
- Bert base uncased finetuned on MRPC dataset
- Bert base uncased fintuned on STS-B dataset

**Features**

- **BERT** tokenizes the input sequence adding a [CLS] token for classification tasks and [SEP] tokens to mark the end of each sentence.
- Each token is encoded with its corresponding embedding vector and positional encoding vectors are added.
- Sentence embeddings are added to differentiate between the sentences if input comes in pairs.
- Attention mask is added to define which tokens are real and which are padding tokens.

**Results**

**Fine-tuned BERT Training Results**

<table>
<thead>
<tr>
<th>Model</th>
<th>N components</th>
<th>Top 5 Accuracy</th>
<th>Explained Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>95</td>
<td>0.966</td>
<td>0.974</td>
</tr>
<tr>
<td>Finetuned on MRPC</td>
<td>0.858</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bert base</td>
<td>113</td>
<td>0.988</td>
<td>0.964</td>
</tr>
<tr>
<td>Finetuned on MRPC</td>
<td>107</td>
<td>0.888</td>
<td>0.967</td>
</tr>
</tbody>
</table>

**Discussion**

- The finetuned semantic similarity BERT models surprisingly performed worse on the semantic search task than both the baseline and bert base.
- This may be because the training sets were tight paraphrases whereas the validation set was composed of very loose paraphrases.
- Results from 768 to 1000 dim improved performance for the baseline, bert base, and the finetuned on MRPC models, suggesting overfitting.
- Interestingly, performance fluctuations around 20 principal components.

**Future**

- Train on a large dataset of the same type of paraphrases as the validation set.
- Finetuning on different layers of the bert base model, which may capture different semantic and structural meaning.

**References**