Using BERT in downstream tasks

CS 585, Fall 2019

Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst

many slides from Jacob Devlin
stuff from last time

• slides ahead of time?
• can you repeat questions asked during class?
• project??? (milestone 1 due Oct 24)
• HW2??? (due Oct 18)
• midterm??? (Oct 31)
rough details about midterm

• two practice exams on Piazza (solutions soon)
  • many problems are on topics we haven’t covered so don’t worry about knowing everything

• topics that can definitely be on the midterm:
  • naive Bayes
  • ngram LMs + smoothing
  • word embeddings (e.g., word2vec)
  • neural networks (e.g., backprop, training setup)
  • fixed-window / RNN LMs
  • Transformers
  • Transfer learning (e.g., ELMo / BERT)
  • sequence labeling (next week)
Masked LM

- **Solution**: Mask out $k\%$ of the input words, and then predict the masked words
  - We always use $k = 15\%$

What are the pros and cons of increasing $k$?
- Use 30,000 WordPiece vocabulary on input.
- Each token is sum of three embeddings
- Single sequence is much more efficient.
Model Architecture

Transformer encoder

- Multi-headed self attention
  - Models context
- Feed-forward layers
  - Computes non-linear hierarchical features
- Layer norm and residuals
  - Makes training deep networks healthy
- Positional embeddings
  - Allows model to learn relative positioning
Empirical advantages of Transformer vs. LSTM:

1. Self-attention == no locality bias
   - Long-distance context has "equal opportunity"

2. Single multiplication per layer == efficiency on TPU
   - Effective batch size is number of words, not sequences

What are they?
Empirical advantages of Transformer vs. LSTM:

1. Self-attention == no locality bias
   - Long-distance context has “equal opportunity”

2. Single multiplication per layer == efficiency on TPU
   - Effective batch size is number of words, not sequences
Model Details

- **Data**: Wikipedia (2.5B words) + BookCorpus (800M words)
- **Batch Size**: 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
- **Training Time**: 1M steps (~40 epochs)
- **Optimizer**: AdamW, 1e-4 learning rate, linear decay
- **BERT-Base**: 12-layer, 768-hidden, 12-head
- **BERT-Large**: 24-layer, 1024-hidden, 16-head
- **Trained on 4x4 or 8x8 TPU slice for 4 days**
Fine-Tuning Procedure

Pre-training

Fine-Tuning

Masked Sentence A

Masked Sentence B

Unlabeled Sentence A and B Pair

Question

Paragraph

Question Answer Pair
Fine-Tuning Procedure

(a) Sentence Pair Classification Tasks: MNLi, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER
GLUE Results

<table>
<thead>
<tr>
<th>System</th>
<th>MNLI-(m/mm)</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST-2</th>
<th>CoLA</th>
<th>STS-B</th>
<th>MRPC</th>
<th>RTE</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>392k</td>
<td>363k</td>
<td>108k</td>
<td>67k</td>
<td>8.5k</td>
<td>5.7k</td>
<td>3.5k</td>
<td>2.5k</td>
<td></td>
</tr>
<tr>
<td>Pre-OpenAI SOTA</td>
<td>80.6/80.1</td>
<td>66.1</td>
<td>82.3</td>
<td>93.2</td>
<td>35.0</td>
<td>81.0</td>
<td>86.0</td>
<td>61.7</td>
<td>74.0</td>
</tr>
<tr>
<td>BiLSTM+ELMo+Attn</td>
<td>76.4/76.1</td>
<td>64.8</td>
<td>79.9</td>
<td>90.4</td>
<td>36.0</td>
<td>73.3</td>
<td>84.9</td>
<td>56.8</td>
<td>71.0</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>82.1/81.4</td>
<td>70.3</td>
<td>88.1</td>
<td>91.3</td>
<td>45.4</td>
<td>80.0</td>
<td>82.3</td>
<td>56.0</td>
<td>75.2</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;BASE&lt;/sub&gt;</td>
<td>84.6/83.4</td>
<td>71.2</td>
<td>90.1</td>
<td>93.5</td>
<td>52.1</td>
<td>85.8</td>
<td>88.9</td>
<td>66.4</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;LARGE&lt;/sub&gt;</td>
<td>&lt;span style=&quot;text-align: center;&quot;&gt;86.7/85.9&lt;/span&gt;</td>
<td>72.1</td>
<td>91.1</td>
<td>94.9</td>
<td>60.5</td>
<td>86.5</td>
<td>89.3</td>
<td>70.1</td>
<td>81.9</td>
</tr>
</tbody>
</table>

### MultiNLI

**Premise:** Hills and mountains are especially sanctified in Jainism.

**Hypothesis:** Jainism hates nature.

**Label:** Contradiction

### CoLa

**Sentence:** The wagon rumbled down the road.

**Label:** Acceptable

**Sentence:** The car honked down the road.

**Label:** Unacceptable
A girl is going across a set of monkey bars. She
(i) jumps up across the monkey bars.
(ii) struggles onto the bars to grab her head.
(iii) gets to the end and stands on a wooden plank.
(iv) jumps up and does a back flip.

- Run each Premise + Ending through BERT.
- Produce logit for each pair on token 0 ([CLS])

\[ P_i = \frac{e^{V \cdot C_i}}{\sum_{j=1}^{4} e^{V \cdot C_j}} \]
Effect of Pre-training Task

- Masked LM (compared to left-to-right LM) is very important on some tasks, Next Sentence Prediction is important on other tasks.
- Left-to-right model does very poorly on word-level task (SQuAD), although this is mitigated by BiLSTM
Effect of Directionality and Training Time

- Masked LM takes slightly longer to converge because we only predict 15% instead of 100%
- But absolute results are much better almost immediately
Effect of Model Size

- Big models help a lot
- Going from 110M -> 340M params helps even on datasets with 3,600 labeled examples
- Improvements have not asymptoted
Multilingual BERT

- Trained single model on 104 languages from Wikipedia. Shared 110k WordPiece vocabulary.

<table>
<thead>
<tr>
<th>System</th>
<th>English</th>
<th>Chinese</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>XNLI Baseline - Translate Train</td>
<td>73.7</td>
<td>67.0</td>
<td>68.8</td>
</tr>
<tr>
<td>XNLI Baseline - Translate Test</td>
<td>73.7</td>
<td>68.4</td>
<td>70.7</td>
</tr>
<tr>
<td>BERT - Translate Train</td>
<td>81.9</td>
<td>76.6</td>
<td>77.8</td>
</tr>
<tr>
<td>BERT - Translate Test</td>
<td>81.9</td>
<td>70.1</td>
<td>74.9</td>
</tr>
<tr>
<td>BERT - Zero Shot</td>
<td>81.9</td>
<td>63.8</td>
<td>74.3</td>
</tr>
</tbody>
</table>

- XNLI is MultiNLI translated into multiple languages.
- Always evaluate on human-translated Test.
- **Translate Train**: MT English Train into Foreign, then fine-tune.
- **Translate Test**: MT Foreign Test into English, use English model.
- **Zero Shot**: Use Foreign test on English model.
Common Questions

● Why did no one think of this before?
● Better question: Why wasn’t contextual pre-training popular before 2018 with ELMo?
● Good results on pre-training is >1,000x to 100,000 more expensive than supervised training.
  ○ E.g., 10x-100x bigger model trained for 100x-1,000x as many steps.
  ○ Imagine it’s 2013: Well-tuned 2-layer, 512-dim LSTM sentiment analysis gets 80% accuracy, training for 8 hours.
  ○ Pre-train LM on same architecture for a week, get 80.5%.
  ○ Conference reviewers: “Who would do something so expensive for such a small gain?”
Common Questions

- The model must be learning more than “contextual embeddings”
- Alternate interpretation: Predicting missing words (or next words) requires learning many types of language understanding features.
  - syntax, semantics, pragmatics, coreference, etc.
- Implication: Pre-trained model is much bigger than it needs to be to solve specific task
- Task-specific model distillation works very well
Common Questions

• Is modeling “solved” in NLP? I.e., is there a reason to come up with novel model architectures?
  ○ But that’s the most fun part of NLP research :(

• Maybe yes, for now, on some tasks, like SQuAD-style QA.
  ○ At least using the same deep learning “lego blocks”

• Examples of NLP models that are not “solved”:
  ○ Models that minimize total training cost vs. accuracy on modern hardware
  ○ Models that are very parameter efficient (e.g., for mobile deployment)
  ○ Models that represent knowledge/context in latent space
  ○ Models that represent structured data (e.g., knowledge graph)
  ○ Models that jointly represent vision and language
Common Questions

● Personal belief: Near-term improvements in NLP will be mostly about making clever use of “free” data.
  ○ Unsupervised vs. semi-supervised vs. synthetic supervised is somewhat arbitrary.
  ○ “Data I can get a lot of without paying anyone” vs. “Data I have to pay people to create” is more pragmatic distinction.

● No less “prestigious” than modeling papers:
  ○ *Phrase-Based & Neural Unsupervised Machine Translation*, Facebook AI Research, EMNLP 2018 Best Paper
Conclusions

- Empirical results from BERT are great, but biggest impact on the field is:
- With pre-training, bigger == better, without clear limits (so far).
- Unclear if adding things on top of BERT really helps by very much.
  - Good for people and companies building NLP systems.
  - Not necessary a “good thing” for researchers, but important.
draw stuff
masked transformer exercise