Extending Answer Prediction for Deep Bidirectional Transformers

David Zhao, Lauren Zhu, Boyang Dun
{dzhao0, laurenz, bodun} @ stanford.edu

Motivation

- The Stanford Question Answering Dataset (SQuAD) is one of the most comprehensive question answering datasets. SQuAD 2.0 builds upon SQuAD 1.1's questions with variable length candidate answers by adding unanswerable questions
- Architectures incorporating Bidirectional Encoder Representations for Transformers (BERT) are reaching near-human performance on SQuAD 2.0
- Vanilla BERT uses a single dense layer on the output contextual representations to predict anwer spans for SQuAD 2.0

https://vignette.wikia.nocookie.net/muppet/images/c/c1/Bert_smile.png/revision/latest?cb=20150630172259

Problem Definition

Given a passage context and a question, provide the correct answer (which can be none) to the question based on the context.

Solution

We investigate extensions to Deep Bidirectional Transformers with models including Pointer-Net, Dynamic Pointing Decoder (DD), and Dynamic Chunk Reader (DCR). We train these hybrid models on the Stanford Question Answering Dataset to learn how to answer questions.

Dataset

Dataset

- We work with SQuAD 2.0, a dataset created by the NLP group at Stanford. As an extension from version 1.1, SQuAD 2.0 contains around 150,000 tuples of question, passage (Wikipedia articles) and answer
- Roughly 1/3 of the question/passage pairs have no answer
- Rest have continuous span of passage text as the answer
- We use a modified 92/4/4 train/dev/test split derived from the official SQuAD dataset

SQuAD2.0 The Stanford Question Answering Dataset

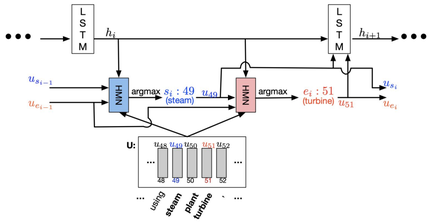
Training example:

Question: What population descended from the Vlachs?
Answer: Moravian Wallachia
Context: "... Romance-speaking Vlachs who migrated into the region ... The population of *Moravian Wallachia* also descend of this population."

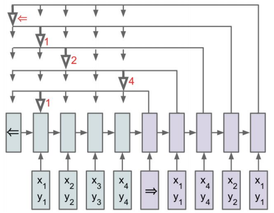
Model Architecture

Dynamic Decoder

- The Dynamic Pointing Decoder (DD) iteratively selects and improves an answer span based on the BERT hidden states
- It passes previous estimates of start and end token indices along with the previous hidden state as input to a LSTM, whose hidden state of the LSTM is passed through Hidden Maxout Networks (HMNs) to update the predictions for the start and end answer token



Pointer Net

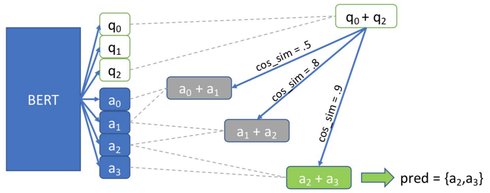


- Pointer-Net is an RNN can generate dynamically-sized outputs
- At each timestep, it outputs a softmax distribution over the length of input
- We draw two outputs: logits for predicted start index, and logits for predicted end index. If predicted start index is beyond end index, model predicts "no answer"

Left: Blue indicates RNN used to encode input sequence, purple indicates network that at each step generates softmax over inputs→ probability of each word being next output token

Answer Chunk Ranker

- The Answer Chunk Ranker (ACR) takes in BERT hidden states and uses a ranking and thresholding module to select the top-scoring span
- We exhaustively generate all answer spans up to a max length *N*, and rank each answer span by its cosine similarity with the concatenation of start and end hidden states of the question



- We use an iterative greedy approach to generate the start and end logits used in training, and implement a variance-based thresholding algorithm for no-answer generation

Results

Model Name	Dev F1 Score	Dev EM Score	Test F1 Score	Test EM Score
Baseline BiDAF	58.75	55.42	59.920	56.298
Fine-tuned BERT (vanilla)	72.898	69.826	-	-
PointerNet-BERT	52.122	52.122	-	-
ACR-BERT	20.666	11.566	-	-
DD-BERT	75.884	72.771	75.035	71.784

- DD-BERT was our best performing model, while ACR-BERT was our worst performing.
- Our fine-tuned BERT and DD-BERT models both outperformed the class-provided baseline.
- PointerNet-BERT F1 and EM scores were the same because it predicted "no answer" on all questions
- ACR-BERT exact match score greatly differs from its F1 score, unlike the other 4 models which track closely
- ACR-BERT's performance suffered because of low performance on the dev set (7.86 EM and F1)

Example Analysis

The DD was our most successful experiment, improving from vanilla BERT by a few points on both F1 and EM.

Consider the following passage and respective question examples. We compare DD-BERT and vanilla pre-trained BERT output.

Context: A particularly simple example of a probabilistic test is the Fermat primality test, which relies on the fact (Fermat's little theorem) that np≡n (mod p) for any n if p is a prime number... More powerful extensions of the Fermat primality test, such as the Baillie-PSW, Miller-Rabin, and Solovay-Strassen tests, are guaranteed to fail at least some of the time when applied to a composite number.

Question: What does the Carmichael primality test depend on?
Ground Truth Answers: No Answer
DD-BERT Prediction: No Answer
Vanilla BERT Prediction: the fact (Fermat's little theorem) that np≡n (mod p) for any n if p is a prime number

- Notice that the correct answer is "No Answer" because a "Carmichael" primality test does not exist—only a "Fermat" one.
- The vanilla BERT model likely hit a local maximum and made an assumption that Carmichael is truly a primality test.
- For DD-BERT, the contextual representations for "Carmichael primality test" are iteratively applied to the context, updating the start and end token predictions. Because each token is individually but sequentially applied to the HMN, the lack of the token sequence "Carmichael primality" in the context warranted the question unanswerable.
- While DD-BERT does improve from vanilla BERT by more closely examining the question's direct relation to tokens in the passage, it still misses the more challenging task of understanding the exact meaning of the question and therefore what to look for or avoid in the passage.

Future

Hyperparameters
Perform a more extensive hyperparameter search for training

No Answer Prediction
Specifically tune PointerNet threshold for classifying logit predictions as "no answer"

Longer Training Time
Longer training time should result in better predictions on PointerNet as logit values systematically shift upwards to create predictions that aren't only "no answer"

References

1. Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. arXiv e-prints, page arXiv:1806.03822, Jun 2018.
2. Jacob, Devlin. Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CORR, abs/1810.04805, 2018.
3. Laddie132. Match-lstm. https://github.com/laddie132/Match-LSTM
4. Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic Coattention Networks for Question Answering. CoRR, abs/1611.01604, 2016.