Stanford University

# BERT for Question Answering on SQuAD 2.0

Zhaozhuo Xu, Yuwen Zhang

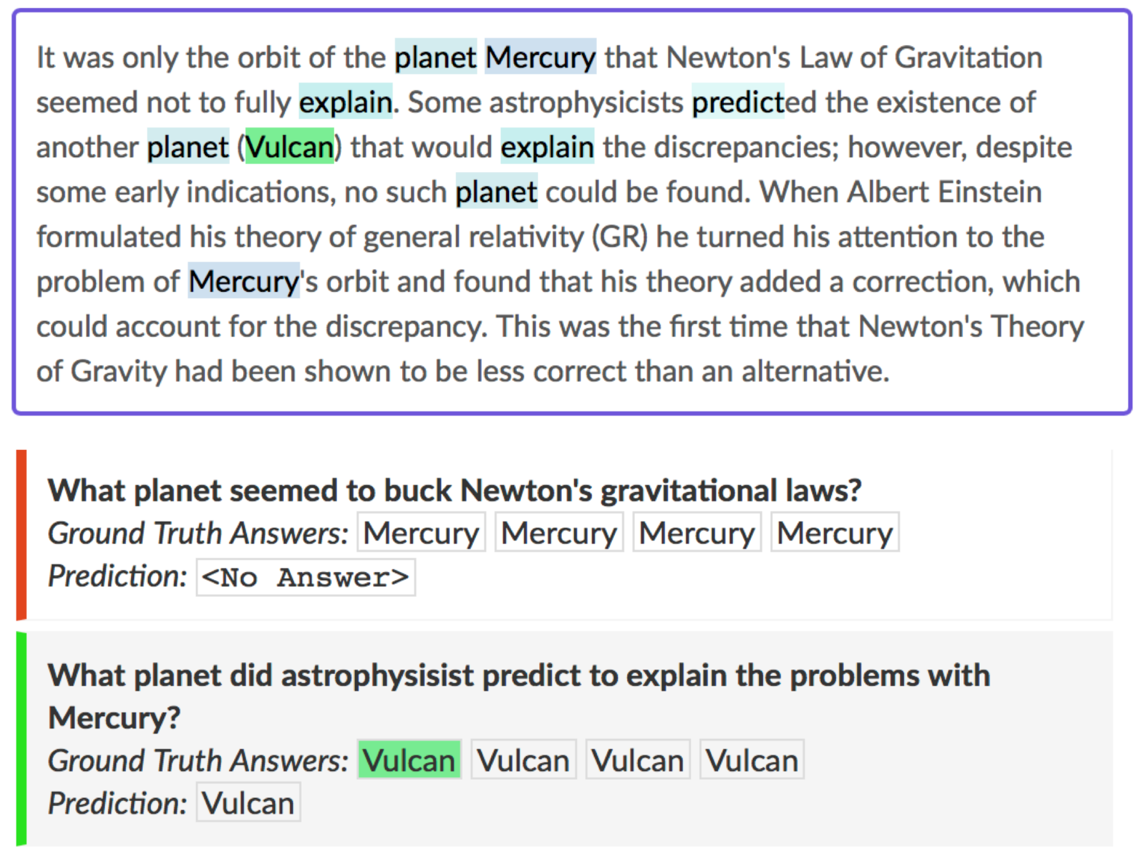
CS 224N

## Problem

- Machine reading comprehension and question answering is an essential task in natural language processing. It is always challenging since it requires a comprehensive understanding of natural languages and the ability to do further inference and reasoning.
- Recently, **Pre-trained Contextual Embeddings (PCE)** models like **ELMo** and **BERT** have attracted lots of attention due to their great performance in a wide range of NLP tasks.
- In this project, we picked up BERT model and tried to fine-tune it with **additional task-specific layers** to improve its performance on **Stanford Question Answering Dataset (SQuAD 2.0)**

## Data

- We used **Stanford Question Answering Dataset (SQuAD 2.0)** to train and evaluate our models.
- Samples in this dataset include **(question, answer, context paragraph)** tuples.
- The paragraphs are from Wikipedia. The questions and answers were crowdsourced using Amazon Mechanical Turk.
- We have around **150k** questions in total, and roughly half of the questions are not answerable.
- If a question is answerable, the answer is guaranteed to be a continuous span in the context paragraph.

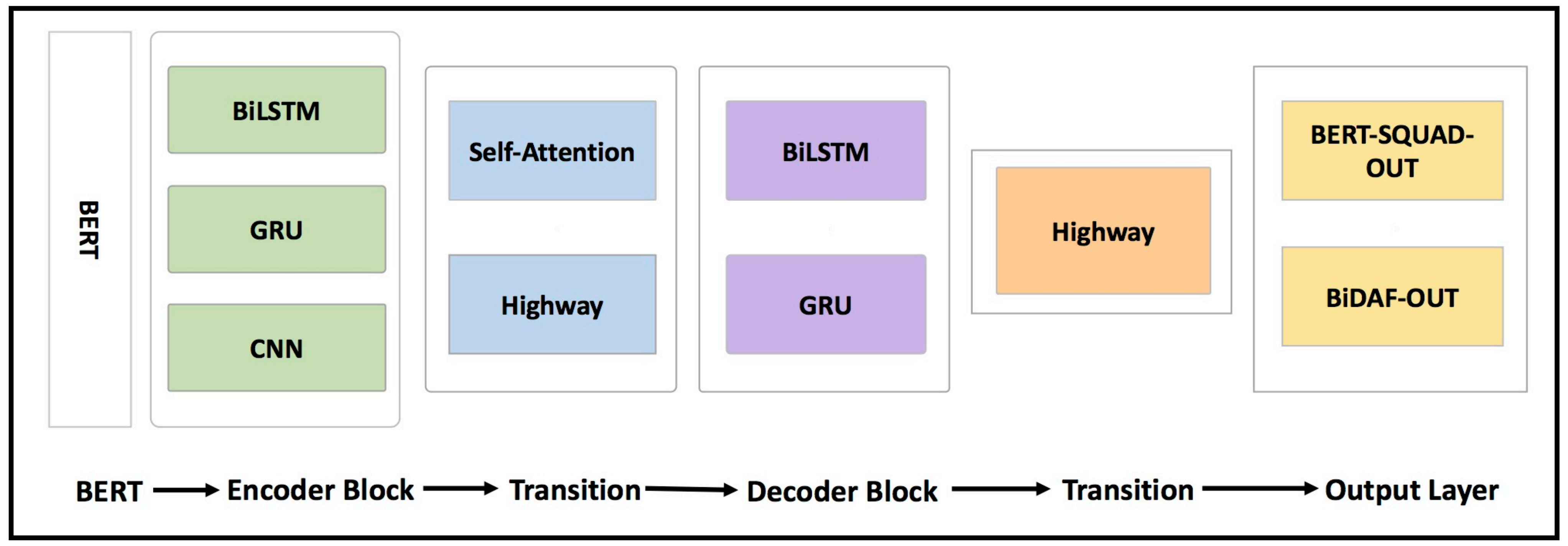


It was only the orbit of the planet Mercury that Newton's Law of Gravitation seemed not to fully explain. Some astrophysicists predicted the existence of another planet (Vulcan) that would explain the discrepancies; however, despite some early indications, no such planet could be found. When Albert Einstein formulated his theory of general relativity (GR) he turned his attention to the problem of Mercury's orbit and found that his theory added a correction, which could account for the discrepancy. This was the first time that Newton's Theory of Gravity had been shown to be less correct than an alternative.

**What planet seemed to buck Newton's gravitational laws?**
*Ground Truth Answers:* Mercury Mercury Mercury Mercury
*Prediction:* `<No Answer>`

**What planet did astrophysisist predict to explain the problems with Mercury?**
*Ground Truth Answers:* Vulcan Vulcan Vulcan Vulcan
*Prediction:* Vulcan

## Approach


BERT
BiLSTM
GRU
CNN
Self-Attention
Highway
BiLSTM
GRU
Highway
BERT-SQUAD-OUT
BiDAF-OUT
BERT → Encoder Block → Transition → Decoder Block → Transition → Output Layer

✓ **Our main idea** is to add an encoder-decoder architecture on top of the BERT model. This idea comes from the computer vision area. For multi-view synthesis task, we can use a general auto-encoder to generate the sketch of other views and an additional auto-encoder for texture level reconstruction.

✓ **Modules on Top of BERT**

| | |
|---|---|
| **BERT** | **BERT** is a multi-layer bidirectional encoder. By training deep transformers on a carefully designed bidirectional language modeling task, we can get the pre-trained BERT representations that we are going to fine-tune later with additional output layers. |
| **Encoder/Decoder** | **RNN Encoder (Bi-LSTM, GRU):** try to integrate temporal dependencies between time-steps of the output tokenized sequence better.<br>**CNN Encoder:** 2D-Convolution on the dimension of seq_len and hidden_state. Extract the relationship of nearby word embeddings in the sequence. |
| **Self-Attention** | On top of CNN or LSTM to introduce extra source of information to guide the extraction of sentence embeddings. Each position of the output token attends to all positions up to and including that position. This can help for better interpreting the inference between different positions in the output sequence. |
| **Highway** | Optimization of networks with virtually arbitrary depth. By applying a gating mechanism, a neural network can have paths along which information can flow across several layers without attenuation. It serves as multi-layer state transitions in RNN to allow the network to adaptively copy or transform representations. |
| **Output** | **BERT Output Layer:** A simple linear output layer<br>**BiDAF Output Layer:** the modeling layer + the output layer discussed in BiDAF model. |

## Results and Analysis

✓ **Results Summary**

| ID | **Architecture on Top of BERT** | **F1** | **EM** |
|---|---|---|---|
| 1 | BERT-base PyTorch Implementation | 76.70 | 73.85 |
| 2 | BERT-base Tensorflow Implementation | 76.07 | 72.80 |
| 3 | GRU Encoder + Self-attention + GRU Decoder + BERT-SQUAD-Out | 73.59 | 69.87 |
| 4 | BiLSTM Encoder + BiDAF-Out | 76.37 | 73.05 |
| 5 | CNN Encoder +Self-attention +BERT-SQUAD-Out | 76.49 | 73.23 |
| 6 | CNN Encoder + BERT-SQUAD-Out | 76.56 | 73.64 |
| 7 | GRU Encoder + GRU Decoder + BERT-SQUAD-Out | 76.85 | 73.77 |
| 8 | CNN Encoder + BiLSTM Decoder + Highway + BERT-SQUAD-Out | 77.07 | 73.87 |
| 9 | BiLSTM Encoder + Highway + BERT-SQUAD-Out | 77.41 | 74.32 |
| 10 | BiLSTM Encoder + Highway + BiLSTM Decoder + BERT-SQUAD-Out | 77.66 | 74.87 |
| 11 | BiLSTM Encoder + BiLSTM Decoder + Highway + BERT-SQUAD-Out | **77.96** | **74.98** |
| 12 | Ensemble of 11 and 7 | **78.35** | **75.60** |
| 13 | Ensemble of 11 and 7 and BERT large case model | **79.44** | **76.966** |

Table 1: F1 and EM scores for different architectures (All our implementations are done in PyTorch).

✓ **Error Analysis Example**

| Questions | **model 1** | **model 3** | **model 11** |
|---|---|---|---|
| Who made fun of the Latin language? | No Answer | Geoffrey Chaucer | No Answer |
| Who led Issacs troops to Cyprus? | No Answer | Guy de Lusignan | No Answer |
| Who began a program of church reform in the 1100s? | No Answer | the dukes | No Answer |

## Conclusion

- We added several components on top of the BERT model as task-specific layers and analyzed their performance compared to BERT baseline model in great details.
- Our best model so far implements **BiLSTM Encoder + BiLSTM Decoder + Highway + BERT-SQUAD-Out** as the output architecture on BERT uncased base model, and it achieves an F1 score of **77.96** on the Dev set.
- With ensemble technique, we finally achieved an F1 score of **79.44** on the Dev Set and **77.827** on the Test Set, ranked **12th** on the 224N leaderboard.

## Reference

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] GitHub. https://github.com/huggingface/pytorch-pretrained-BERT, 2018.