AI2100 #47

Explaining answers given by neural question answering systems

AI2100 Final Report

Paper ID 47

Abstract

Question answering has been a fundamental challenge in NLP with vast applications. State of the art neural question answering systems are able to beat humans at this task when evaluated on carefully prepared datasets. However it is difficult to explain in human comprehensible manner, how these models reach the answers that they do. Such an explanation is often important in high stakes situations such as healthcare or business applications. Hence, we explored various explainability methods in the context of neural question answering through this project and verified some of their results. We also report our findings and frame adversarial examples to support our claims.

1. Approach

We analysed BERT model for the task of RCQA using integrated gradients. We fine tuned BERT-base-uncased for 3 epochs on SQuAD-v1 and saved the fine-tuned model. To understand what the fine-tuned BERT is looking at, and to explain the functionality of its layers, we propose to use the embeddings that are produced after each layer. A few previous works look at attention scores produced by BERT to conclude facts about interpretability. But, more recent work [1] shows that attention, atleast in some cases, is not suited for explainability. Hence we chose to work on the embeddings which carry some information about what BERT has learnt, from each layer. We analyse them using two approaches:

1. Visualisations of integrated gradients of embeddings, after each layer
2. t-SNE visualisations of the embeddings, after each layer.

In the following section, we verify the results which previous works have found. Also, we outline some of our findings for the fine tuned model. Using Integrated gradient technique, we came to the conclusion that Bert model uses certain heuristics to answer question. To further verify this claim, we built adversarial examples on which these heuristics would not work. We found that Bert model did fail on these adversarial examples.

2. Results

2.1. Verification of some of the previous works

1. [2] claims that initial layers of BERT focus on question words in the passage. In the later layers, the focus on question words decreases, and more focus is on supporting words that surround the answer. Their approach uses Integrated Gradients. Proceeding on similar lines, we confirm their claims.

We calculated the attribution scores for embeddings of words after each layer using integrated gradients approach. We leveraged the captum library to achieve this. We visualised these using the inbuilt-captum functionality, as well as by plotting the heatmaps.

Consider the context-question pair given below.
Context: The Panthers finished the regular season with a 15–1 record, and quarterback Cam Newton was named the NFL Most Valuable Player (MVP). They defeated the Arizona Cardinals 49–15 in the NFC Championship Game and advanced to their second Super Bowl appearance since the franchise was founded in 1995. The Broncos finished the regular season with a 12–4 record, and denied the New England Patriots a chance to defend their title from Super Bowl XLIX by defeating them 20–18 in the AFC Championship Game. They joined the Patriots, Dallas Cowboys, and Pittsburgh Steelers as one of four teams that have made eight appearances in the Super Bowl. The 2020 regular season win/loss ratio for the Michigan Vikings was 656.
Question: How many appearances have the Broncos made in the super bowl?
Predicted Answer: eight

We use captum library to plot our results. We indeed get similar results 2 which the paper claims. Initial layers focus more on question words in context. Later layers, try to look

AI2100 #47

at supporting context words. Finally, the model focuses on the answer in the last layers.

2. [3] claims that initial layers look at the semantic meanings of each word, and subsequent layers look at relations between the words in context. The final layers filter out the relevant information, and come up with the answer. Their approach uses PCA visualisation of embeddings. Proceeding on similar lines, we confirm their claims.

Consider the context-question pair given below.
Context: Alfred is a white cat and likes apples. Bobby is a black dog which loves mangoes.
Question: What does Alfred like?
Predicted Answer: Apples

We perform t-SNE visualisations of embeddings after each layer, instead of PCA, in order to preserve more information, which might have possibly lost during PCA. We indeed get similar results 1 which the paper claims. Initial layers look at semantic meanings of words (words like white, black or loves, like, or cat, dog occur together). Later layers, learn relations in context (white, cat, black, dog become together). Finally, all the important information is filtered out (apples, is reasonably seperated).

2.2. Some of our (interesting) findings

Analysing the heat-maps for different examples, we observed few patterns. First was that initially the model was focusing greatly on question words like "Who, What, When, Where". The model was also focusing on possible answers to such nouns and numbers in the query.

Consider the context-question pair given below 3
Context: 93 players have been awarded the Fewest Prized Team distinction for the Champ Bowl.
Question: How many players have been awarded the Fewest Prized Team distinction for the Super Bowl?

Second was that as mentioned previously, the model was focusing on words which were same in the question and the context. Then the model was looking at neighbours of the words it focused on initially, as mentioned above. (Look at section 2.1).

Based on these observations, we claim that the Bert model uses heuristics to predict the start and end tokens. These heuristics work as follows:

1. Recognize the question type by focusing on the question words. Focus on candidate answers.

2. Focus on words that are same in both the question and the answer.

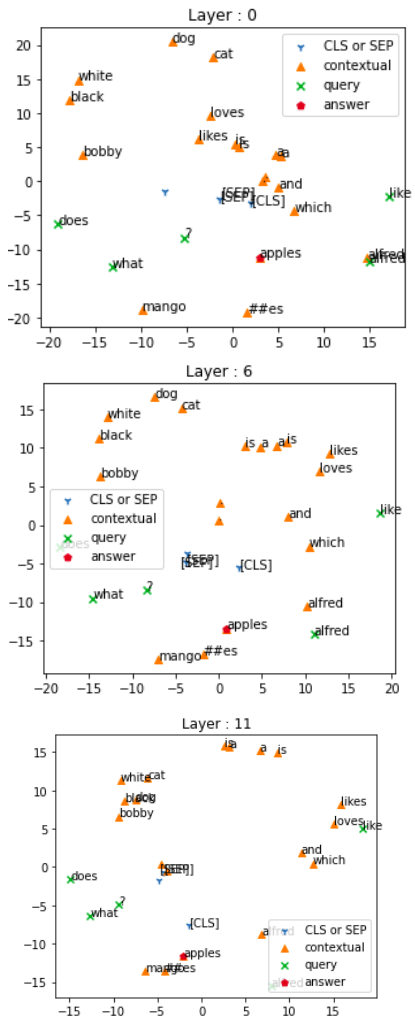

Figure 1. t-SNE visualisations of word embeddings after a layer.

3. Look at neighbouring words of the words on which we focused in the above steps and predict start and end tokens from these.

How exactly the model picks start and end token during heuristic 3 is still a mystery.

2.3. Verification of the claims using adversarial examples

We constructed adversarial examples by assuming that the model uses these heuristics. We expected that the model to fail on these examples and that is exactly what happened.

To test heuristic 1, consider the context-question pair given below
Context: Yes, a person can jump from a hill and still be alive.
Question: After jumping from a hill, can a person be alive?
Predicted Answer: still be alive

Figure 2. Visualisation of Integrated gradients of word embedding after a layer.

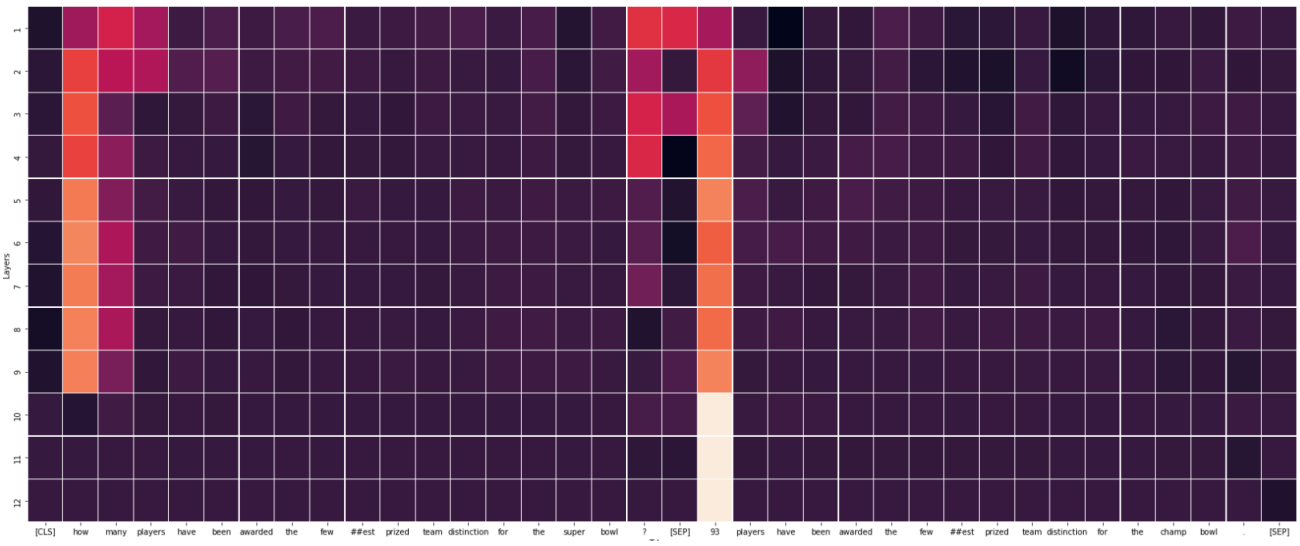

Figure 3. Example 1

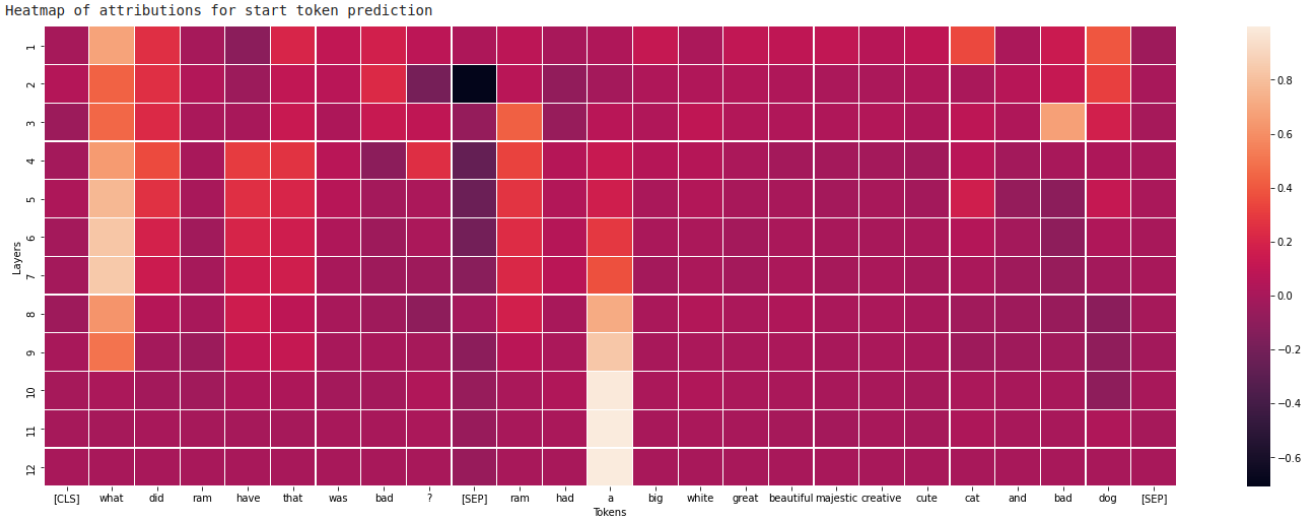

Figure 4. Example

Clearly, the model fails here. To test heuristic 2, we constructed examples where there were no similar words or where there were more than one similar word.

Consider the context-question pairs given below
Context: Trump has a Trump card having his photo.
Question: What has Trumpś photo?
Predicted Answer: Trump card
Context: Ram was drawing with pencils. Teacher was picking her nose with pencil. Ram observed that pencils can be used to pick nose.
Question: What was Ram's conclusion?
Predicted Answer: None

In the last example, if "conclusion" is replaced with "observation" in the query, the model gives correct output. This further shows the necessity of same words. This also makes the model brittle and easy to fool because there is a lack of language understanding.

To test heuristic 3, we constructed an example in which the correct answer was much farther than words on which model is focusing using heuristic one and two.

Consider the context-question pair given below
Context: Ram had a big white great beautiful majestic creative cute cat and bad dog.
Question: What did Ram have that was bad?
Predicted Answer: a big white great beautiful majestic creative cute cat and bad dog

The heat-map for the last example shows us the full story of how the model predicts the answer. See figure 4. First the model focuses on "cat" and "dog" as they are candidate answers for the question word "what". The model also focuses on the word "what". Then the model finds the word "Ram" to be same in both the question and the context. Thus it focuses on "Ram" in the context. Then it looks at it's neighbouring word: "a" and predicts it for start token. The model also focuses on the word "bad" as it is also same in context and question. However, in step 4 of the heuristics, the model chooses to not "go ahead" with the word "bad".

3. Contributions

1. **AI20BTECH11004**: Section 2.2 in preliminary report, section 3 in mid-term report, code for captum in mid-term submission, t-SNE code in final report, minor change in bert-fine tune to save models.
2. **AI20BTECH11006**: Section 2.1 in preliminary report, section 3 in mid-term report, code for captum in mid-term submission, t-SNE code in final report, minor change in bert-fine tune to save models.
3. **AI20BTECH11011**: Abstract, Section 1, Conclusion in preliminary report, Section 2 in mid-term report, bert-fine-tune code in mid-term submission, Section 2.2, 2.3 in final report.
4. **AI20BTECH11015**: Section 2.1 in preliminary report, Section 1 in mid-term report, bert-fine-tune code in mid-term submission, prepared question-ground truth dataset, Section 2.1 in final report.
5. **AI20BTECH11027**: Section 2.2 in preliminary report, Section 1 in mid-term report, bert-fine-tune code in mid-term submission, Captum heat map visualization code in final-report submission, Abstract, Section 1 and 2.1 in final report.

References

[1] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019. 1
[2] Sahana Ramnath, Preksha Nema, Deep Sahni, and Mitesh M. Khapra. Towards interpreting BERT for reading comprehension based QA. *CoRR*, abs/2010.08983, 2020. 1
[3] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. How does BERT answer questions? A layer-wise analysis of transformer representations. *CoRR*, abs/1909.04925, 2019. 2