

Towards Interpreting BERT for Reading Comprehension Based QA

Sahana Ramnath*, Preksha Nema, Deep Sahni, Mitesh M. Khapra

Robert Bosch Centre for Data Science and AI (RBC-DSAI)

Indian Institute of Technology Madras, Chennai, India

{sahanajich, deep.sahani11}@gmail.com,

{preksha, miteshk}@cse.iitm.ac.in

Abstract

BERT and its variants have achieved state-of-the-art performance in various NLP tasks. Since then, various works have been proposed to analyze the linguistic information being captured in BERT. However, the current works do not provide an insight into how BERT is able to achieve near human-level performance on the task of Reading Comprehension based Question Answering. In this work, we attempt to interpret BERT for RCQA. Since BERT layers do not have predefined roles, we define a layer's role or functionality using Integrated Gradients. Based on the defined roles, we perform a preliminary analysis across all layers. We observed that the initial layers focus on query-passage interaction, whereas later layers focus more on contextual understanding and enhancing the answer prediction. Specifically for quantifier questions (how much/how many), we notice that BERT focuses on confusing words (i.e., on other numerical quantities in the passage) in the later layers, but still manages to predict the answer correctly. The fine-tuning and analysis scripts will be publicly available at <https://github.com/iitmnlp/BERT-Analysis-RCQA>.

1 Introduction

The past decade has witnessed a surge in the development of deep neural network models to solve NLP tasks. Pretrained language models such as ELMO (Peters et al., 2018a), BERT (Devlin et al., 2018), XLNet (Yang et al., 2019) etc. have achieved state-of-the-art results on various NLP tasks. This success motivated various studies to understand how BERT achieves human-level performance on these tasks. Tenney et al. (2019); Peters et al. (2018b) analyze syntactic and semantic roles played by different layers in such models. Clark et al. (2019) specifically analyze BERT's attention

heads for syntactic and linguistic phenomena. Most of these works focus on tasks such as sentiment classification, syntactic/semantic tags prediction, natural language inference, and so on. However, to the best of our knowledge, BERT has not been thoroughly analyzed for complex tasks like RCQA. It is a challenging task because of 1) the large number of parameters and non-linearities in BERT, and 2) the absence of pre-defined roles across layers in BERT as compared to pre-BERT models like BiDAF (Seo et al., 2016) or DCN (Xiong et al., 2016). In this work, we take the first step to identify each layer's role using the attribution method of Integrated Gradients (Sundararajan et al., 2017). We then try to map these roles to the following functions, deemed necessary in pre-BERT models to reach the answer: (i) learn contextual representations for the passage and the question, individually, (ii) attend to information in the passage specific to the question and, (iii) predict the answer.

We perform analysis on the SQuAD (Rajpurkar et al., 2016) and DuoRC (Saha et al., 2018) datasets. We observe that the initial layers primarily focus on question words that are present in the passage. In the later layers, the focus on question words decreases, and more focus is on the supporting words that surround the answer and the predicted answer span. Further, through a focused analysis of quantifier questions (questions that require a numerical entity as the answer), we observe that BERT pays importance to many words similar to the answer (same type, such as numbers) in later layers. We find this intriguing since, even after marking confusing words spread across passage as important, BERT's prediction accuracy is high. We also provide qualitative analysis to demonstrate the above trends.

* now at Google Research, Bangalore

2 Related Work

In the past few years, various large-scale datasets have been proposed for the RCQA task (Nguyen et al., 2016; Joshi et al., 2017; Rajpurkar et al., 2016; Saha et al., 2018) which have led to various deep neural-network (NN) based architectures such as Seo et al. (2016); Dhingra et al. (2016). Additionally, with complex pretraining, models such as Liu et al. (2019); Lan et al. (2019); Devlin et al. (2018) are very close to human-level performance. Due to the large number of parameters and non-linearity of deep NN models, the answer to the question “how did the model arrive at the prediction?”, is not known; hence, they are termed as *blackbox models*. Motivated by this question, there have also been many works that analyze the interpretability of deep NN models on NLP tasks; many of them analyze models based on in-built attention mechanisms (Jain and Wallace, 2019; Serrano and Smith, 2019; Wiegrefe and Pinter, 2019). Further, various attribution methods such as Bach et al. (2015); Sundararajan et al. (2017) have been proposed to analyze them. Tenney et al. (2019) and Peters et al. (2018b) perform a layerwise analysis of BERT and BERT-like models to assign them syntactic and semantic meaning using probing classifiers. Si et al. (2019) question BERT’s working on QA tasks through adversarial attacks, similar to Jia and Liang (2017); Mudrakarta et al. (2018). They point out that BERT is prone to be fooled by such attacks. Unlike these earlier works, we focus on analyzing BERT’s layers specifically for RCQA to understand their QA-specific roles and their behavior on potentially confusing quantifier questions.

3 Experimental Setup

For our BERT analysis, we use the BERT-BASE model, which has 12 Transformer blocks (layers), each with a multi-head self-attention and a feed-forward neural network. We use the official code and pre-trained checkpoints¹ and fine-tune it for two epochs for the SQuAD and DuoRC datasets to achieve F1 scores of 88.73 and 54.80 on their respective dev-splits. We use SQuAD (Rajpurkar et al., 2016) 1.1 with 90k/10k train/dev samples, each with a 100-300 words passage and the SelfRC dataset in DuoRC (Saha et al., 2018) with 60k/13k train/dev samples, each with a 500 (on average)

¹<https://github.com/google-research/bert>

words passage. For each passage, both datasets have a natural language query and answer span in the passage itself.

4 Layer-wise Functionality

As discussed earlier, we aim to understand each BERT layer’s functionality for the RCQA task; we want to identify the passage words that are of primary importance at each layer for the answer. Intuitively, the initial layers should focus on question words, and the latter should zoom in on contextual words that point to the answer. To analyze the above, we use the attribution method Integrated Gradients (Sundararajan et al., 2017) on BERT at a layerwise level.

For a given passage P consisting of n words $[w_1, w_2, \dots, w_n]$, query Q , and model f with θ parameters, answer prediction is modeled as:

$$p(w_s, w_e) = f(w_s, w_e | P, Q, \theta)$$

where w_s, w_e are the predicted answer start and end words or positions.

For any given layer l , the above is equivalent to:

$$p(w_s, w_e) = f_l(w_s, w_e | E_{l-1}(P), E_{l-1}(Q), \theta)$$

where f_l is the forward propagation from layer l to the prediction. $E_l(\cdot)$, is the representation learnt for passage or query words by a given layer l . To elaborate, we consider the network below the layer l as a blackbox which generates *input* representations for layer l . The **Integrated Gradients** for a Model M , a passage word w_i , embedded as $x_i \in \mathbf{R}^L$ is:

$$IG(x_i) = \int_{\alpha=0}^1 \frac{\partial M(\tilde{x} + \alpha(x_i - \tilde{x}))}{\partial x_i} d\alpha$$

where \tilde{x} is a zero vector, that serves as a baseline to measure integrated gradient for w_i . We calculate the integrated gradients at each layer $IG_l(x_i)$ for all passage words w_i using Algorithm 1. We approximate the above integral across 50 uniform samples of $\alpha \in [0, 1]$. We then compute importance scores for each w_i by taking the euclidean norm of $IG(w_i)$ and normalizing it to get a probability distribution I_l over the passage words.

4.1 JSD with top-k retained/removed

We quantify and visualize a layer’s function as its distribution of importance over the passage words I_l . To compute the similarity between any

Algorithm 1 To compute Layer-wise Integrated Gradients for layer l

- 1: $\tilde{p} = 0$ //zero baseline
- 2: $m = 50$
- 3: $G_l(p) = \frac{1}{m} \sum_{k=1}^m \frac{\partial f_l(\tilde{p} + \frac{k}{m}(p - \tilde{p}))}{\partial E_l}$
- 4: $IG_l(p) = [(p - \tilde{p}) \times G_l(p)]$
- 5: // Compute squared norm for each word
- 6: $\tilde{I}_l([w_1, \dots, w_k]) = \|IG_l(p)\| \in \mathbb{R}^k$
- 7: Normalize \tilde{I}_l to a probability distribution I_l

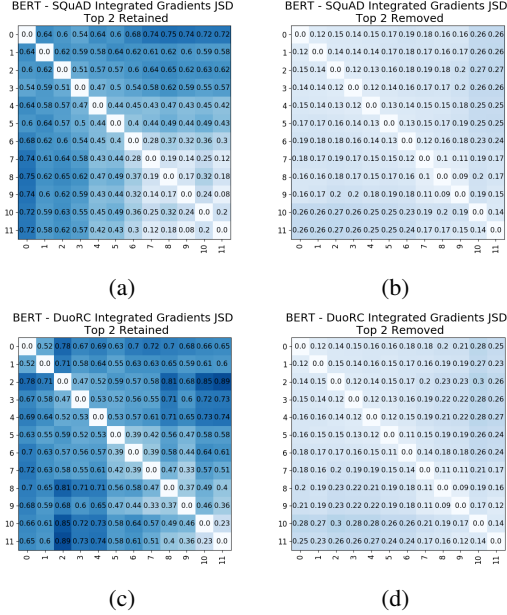


Figure 1: JSD between I_l 's with top-2 items removed/retained (SQuAD - (a), (b), DuoRC - (c), (d))

two layers x, y , we measure the *Jensen-Shannon Divergence (JSD)* between their corresponding importance distributions I_x, I_y . We calculate the JSD scores between every pair of layers in the model and visualize it as a $n_l \times n_l$ heatmap (n_l - number of layers in the model). A higher JSD score corresponds to the two layers being more different. This further means the two layers consider different words as salient. We visualize heatmaps for the dev-splits of SQuAD (Figures 1a, 1b) and DuoRC (Figures 1c, 1d), averaging over 1000 samples in each case.

We analyze the distribution in two parts: (i) we retain only top-k scores in each layer and zero out the rest, which denotes the distribution's head. (ii) we zero the top-k scores in each layer and retain the rest, which denotes the distribution's tail. In either case, we re-normalize to get a probability distribution. When comparing just the top-2 items, we

Layer Name	% answer span	% Q-words	% Contextual Words
Layer 0	26.99	22.94	9.45
Layer 1	26.09	24.35	9.43
Layer 2	29.9	22.41	11.65
Layer 3	30.44	19.55	11.13
Layer 4	30.06	18.33	11.23
Layer 5	30.75	14.71	11.57
Layer 6	31.25	15.33	11.94
Layer 7	32.37	12.29	12.32
Layer 8	30.78	18.91	12.07
Layer 9	34.58	10.21	13.41
Layer 10	34.31	10.56	13.39
Layer 11	34.63	12.0	13.74

Table 1: Semantic statistics of top-5 words - SQuAD

Layer Name	% answer span	% Q-words	% Contextual Words
Layer 0	35.14	17.89	27.53
Layer 1	37.29	18.29	29.88
Layer 2	38.30	19.59	30.05
Layer 3	34.37	18.88	25.83
Layer 4	33.93	20.77	26.20
Layer 5	36.32	16.16	27.97
Layer 6	35.34	15.75	27.05
Layer 7	41.20	10.57	31.12
Layer 8	40.38	8.50	22.16
Layer 9	41.25	8.03	17.9
Layer 10	43.93	5.58	15.85
Layer 11	44.37	6.00	33.74

Table 2: Semantic statistics of top-5 words - DuoRC

see higher values (min 0.08/max 0.72) in heatmap 1a than in heatmap 1b (min 0.09/max 0.26). Similarly, we see higher values (min 0.23/max 0.89) in heatmap 1c than in heatmap 1d (min 0.12/max 0.28). We conclude that a layer's function is reflected in words high up in the importance distribution. As we remove them, we encounter an almost uniform distribution across the less important words. Hence to correctly identify a layer's functionality, we need to focus only on the head (top-k words) and not on the tail.

5 Results and Discussions

5.1 Probing layers: QA functionality

Based on the defined layers' functionality I_l , we try to identify which layers focus more on the question, the context around the answer, etc. We segregate the passage words into three categories: *answer words*, *supporting words*, and *query words*, where supporting words are the words surrounding the answer within a window size of 5. Query words are the question words which appear in the passage. We take the top-5 words marked as important in I_l for any layer l and compute how many words from each of the above-defined categories appear in the top-5 words (results in Tables 1 and 2). We observe

Question: Why was Polonia relegated from the country's top flight in 2013?

Answer: disastrous financial situation

L0	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....	L9	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....
L1	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....	L10	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....
L2	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....	L11	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....

Table 3: Heatmap visualisation of the I_l distribution over BERT's first and last 3 layers, for a sample from SQuAD. The initial layers focus on question specific words and latter focus on supporting words that lead to answer

similar overall trends for both SQuAD and DuoRC. From Column 3, it is evident that the model first tries to identify the part of the passage where the question words are present. As it gets more confident about the answer (Column 2), the question words' importance decreases. From Col. 4, we infer that the layers' contextual role increases from the initial to the final layers.

Qualitative Example: We present a visualization of the top-5 words of the first and last three layers (with respect to I_l) in Table 3 for a sample from SQuAD. We see that all six layers give a high score to the answer span itself ('disastrous', 'situation'). Further, we see that the initial layers 0,1 and 2 are also trying to connect the passage and the query ('relegated', 'because', 'Polonia' get high importance scores). Hence, in this example, we see that the initial layers incorporate interaction between the query and passage. In contrast, the last layers focus on enhancing and verifying the model's prediction.

5.2 Visualizing Word Representations

We now qualitatively analyze the word representations of each layer. We visualize the t-SNE plot for one such passage, question, answer triplet from SQuAD (refer Table 4) in Figures 2, 3. We visualize the answer, supporting words, query words, and special tokens. Note that we have grayed out the other words in the passage. In initial layers (such as layer 0), we observe that similar words such as stop-words, team names, numbers {eight, four}, etc., are close to each other. In Layer 4, the passage, question, and answer come closer to each other. By layer 9, we see that the answer words are segregated from the rest of the words, even though the passage word 'four', which is of the same type as the answer 'eight' (number), is still close to 'eight'. We see more interesting observations yet here: (i)

Passage: the panthers finished the regular season with a 15 – 1 record, ... the broncos ... finished the regular season with a 12 – 4 record. They joined the patriots , dallas cowboys , and pittsburgh steelers as one of teams that have made eight appearances in the super bowl .

Question: How many appearances have the Broncos made in the super bowl?

Table 4: Sample from the dev-split of SQuAD. Blue shows the answer, purple shows the contextual passage words and green shows the query

in later layers, the question words separate from the answer and the supporting words, (ii) Across all 12 layers, embeddings for *four*, *eight* remain very close together, which could have easily led to the model making a wrong prediction. However, the model still predicts the answer 'eight' correctly. We were not able to identify the layer where the distinction between the two confusing answers occurs.

Quantifier questions: For a detailed analysis of quantifier questions like *how many*, *how much* that could have many confusing answers (i.e., numerical words) in the passage, we perform further analysis. Based on our layer-level functionality I_l , we compute the number of words that are numerical quantities in the top-5 words, and the entire passage, and compute their ratio. This represents the ratio of confusing words that are marked as important by each layer. There are 799 and 310 such questions in SQuAD and DuoRC, respectively.

Interestingly, we observe that this ratio *increases* as we go higher up (SQuAD: L0 - 5.6%, L10 - 17.7%, L11 - 15.5%, DuoRC: L0 - 12.9%, L10 - 21.6%, L11 - 22.6%). For the example in Table 4, we observed that in its later layers, BERT gives high importance to the words 'eight', 'four', and

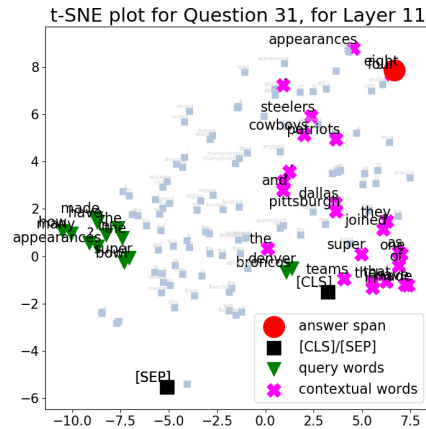
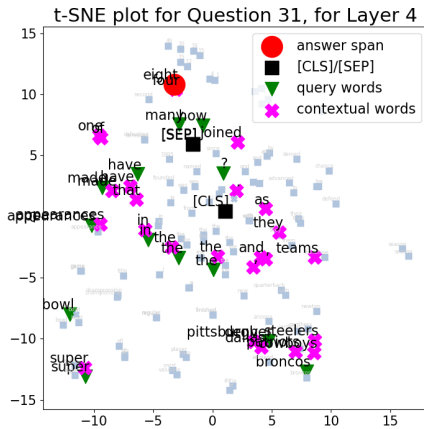
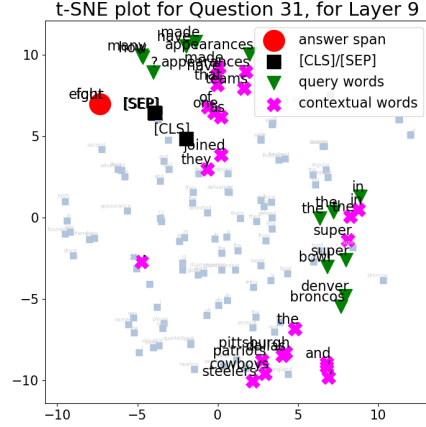
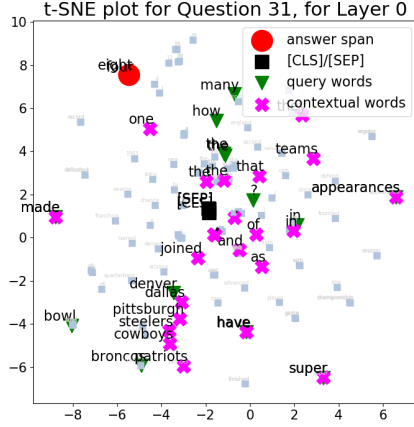


Figure 2: t-SNE plots - word embeddings of layers 0, 4 for the example in Table 4. For layer 0 similar words (e.g., team names, stop words) are close to each other. For intermediate layers like Layer 4, all the contextual, answer and question words intermingle.

Figure 3: t-SNE plots- word embeddings of layers 9, 11 for the example in Table 4. In layers 9-11, the answer *eight* segregates from other words. However, numerical entity *four*, is very close to the answer.

6 Conclusion

‘second’ (numerical quantities), even though the latter are not related or necessary to answer the question. This shows that BERT, in its later layers, distributes its focus over confusing words. However, it *still* manages to predict the correct answer for such questions (87.35% EM for such questions for SQuAD, and 53.5% in DuoRC); BERT also has high confidence in predicting the answer for such questions (86.5% vs 80.4% for quantifier questions with more than one numerical entity in the passage vs non-quantifier questions in SQuAD, 95.2% vs 87.2% in DuoRC). This behavior is very different from the assumed roles a layer might take to answer the question, as it is expected that such words were considered in the initial rather than final layers. This shows the complexity of BERT and the difficulty of interpreting it for the RCQA task.

In this work, we highlight that the lack of predefined roles for layers adds to the difficulty of interpreting highly complex BERT-based models. We first define each layer’s functionality using Integrated Gradients. We present results and analysis to show that BERT is learning some form of passage-query interaction in its initial layers before arriving at the answer. We found the following observations interesting and with a potential to be probed further: (i) why do the question word representations move away from contextual and answer representation in later layers? (ii) If the focus on confusing words increases from the initial to later layers, how does BERT still have a high accuracy? We hope that this work will help the research community interpret BERT for other complex tasks and explore the above open-ended questions.

Acknowledgements

We thank the Department of Computer Science and Engineering, IIT Madras and the Robert Bosch Center for Data Science and Artificial Intelligence, IIT Madras (RBC-DSAI) for providing us compute resources. We thank Google for supporting Preksha Nema contribution through the Google Ph.D. Fellowship programme. We also thank the anonymous reviewers for their valuable and constructive suggestions.

References

- Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. [On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation](#). *PLOS ONE*, 10.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? an analysis of bert’s attention](#). *CoRR*, abs/1906.04341.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Bhuvan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. [Gated-attention readers for text comprehension](#). *arXiv preprint arXiv:1606.01549*.
- Sarthak Jain and Byron C. Wallace. 2019. [Attention is not explanation](#). *CoRR*, abs/1902.10186.
- Robin Jia and Percy Liang. 2017. [Adversarial examples for evaluating reading comprehension systems](#). *CoRR*, abs/1707.07328.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. [Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension](#). *CoRR*, abs/1705.03551.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. [ALBERT: A Lite BERT for Self-supervised Learning of Language Representations](#). *arXiv e-prints*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhare. 2018. [Did the model understand the question?](#) *CoRR*, abs/1805.05492.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. [MS MARCO: A human generated machine reading comprehension dataset](#). *CoRR*, abs/1611.09268.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. [Deep contextualized word representations](#). *arXiv preprint arXiv:1802.05365*.
- Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. [Dissecting contextual word embeddings: Architecture and representation](#). *arXiv preprint arXiv:1808.08949*.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [Squad: 100, 000+ questions for machine comprehension of text](#). *CoRR*, abs/1606.05250.
- Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. [Duorc: Towards complex language understanding with paraphrased reading comprehension](#). *CoRR*, abs/1804.07927.
- Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. [Bidirectional attention flow for machine comprehension](#). *CoRR*, abs/1611.01603.
- Sofia Serrano and Noah A Smith. 2019. [Is attention interpretable?](#) *arXiv preprint arXiv:1906.03731*.
- Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. 2019. [What does bert learn from multiple-choice reading comprehension datasets?](#) *arXiv preprint arXiv:1910.12391*.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. [Axiomatic attribution for deep networks](#). *CoRR*, abs/1703.01365.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [Bert rediscovers the classical nlp pipeline](#). *arXiv preprint arXiv:1905.05950*.
- Sarah Wiegrefe and Yuval Pinter. 2019. [Attention is not not explanation](#). *arXiv preprint arXiv:1908.04626*.
- Caiming Xiong, Victor Zhong, and Richard Socher. 2016. [Dynamic coattention networks for question answering](#). *CoRR*, abs/1611.01604.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in neural information processing systems*, pages 5754–5764.

Layer Name	% common / proper / cardinal nouns	% verbs	% stop words	% adverbs	% adjectives	% punct marks	% words in answer span
Layer 0	49.57	12.92	12.63	2.73	11.63	11.41	26.99
Layer 1	53.65	13.81	10.71	3.13	11.44	8.27	26.09
Layer 2	52.16	14.19	13.71	3.24	12.52	5.25	29.9
Layer 3	49.63	12.98	16.27	2.76	10.97	8.52	30.44
Layer 4	47.99	12.32	19.93	2.87	10.58	7.29	30.06
Layer 5	46.97	12.34	19.35	2.73	9.56	10.29	30.75
Layer 6	49.61	12.13	17.38	2.51	9.74	9.75	31.25
Layer 7	50.43	11.31	16.23	2.61	9.87	10.85	32.37
Layer 8	54.16	11.59	14.59	2.58	11.27	6.94	30.78
Layer 9	53.09	10.11	12.98	2.42	11.01	11.82	34.58
Layer 10	57.8	8.67	12.2	2.11	10.93	9.64	34.31
Layer 11	54.58	8.77	14.57	2.31	10.43	10.63	34.63

Table 5: Part-of-Speech statistics of top-5 words - SQuAD

Layer Name	% common / proper / cardinal nouns	% verbs	% stop words	% adverbs	% adjectives	% punct marks	% words in answer span
Layer 0	55.81	12.63	9.5	1.97	9.56	10.87	35.14
Layer 1	58.1	13.21	8.41	2.16	10.03	8.6	37.29
Layer 2	59.42	13.9	8.67	2.22	10.54	5.61	38.30
Layer 3	55.03	13.61	11.55	2.15	9.54	8.78	34.37
Layer 4	54.43	13.91	12.63	1.97	9.14	8.26	33.93
Layer 5	51.97	13.09	12.58	1.82	8.04	12.79	36.32
Layer 6	54.88	12.35	9.84	1.77	8.45	12.88	35.34
Layer 7	60.12	10.02	9.34	1.8	9.07	9.94	41.20
Layer 8	60.81	8.56	7.64	1.84	9.2	12.33	40.38
Layer 9	60.96	8.84	8.2	1.84	9.24	11.33	41.25
Layer 10	57.43	8.42	10.57	1.81	9.05	13.24	43.93
Layer 11	60.46	9.07	11.06	1.97	9.39	8.65	44.37

Table 6: Part-of-Speech statistics of top-5 words - DuoRC

A Probing layers: POS Tags

Based on the layers’ functionality I_l , we analyze the top-5 important words in each layer on the basis of POS tags. The results can be found in Tables 5 and 6. We note that all 12 layers are majorly focused on entity based words (common nouns, proper nouns and numerical entities). Surprisingly, all layers give approximately 10% of their importance to punctuation marks and stopwords each, the same level of importance that is given to verbs and adjectives. It is worth noting that on average, answer spans in SQuAD on 82.04% entites, and answer spans in DuoRC are 79.78% entities.