How Does BERT Answer Questions? A Layer-Wise Analysis of Transformer Representations

Anonymous Author(s)*

ABSTRACT

Bidirectional Encoder Representations from Transformers (BERT) reach state-of-the-art results in a variety of Natural Language Processing tasks. However, understanding of their internal functioning is still insufficient and unsatisfactory. In order to better understand BERT and other Transformer-based models, we present a layer-wise analysis of BERT's hidden states. Unlike previous research, which mainly focuses on explaining Transformer models by their attention weights, we argue that hidden states contain equally valuable information. Specifically, our analysis focuses on models fine-tuned on the task of Question Answering (QA) as an example of a complex downstream task. We inspect how QA models transform token vectors in order to find the correct answer. To this end, we apply a set of general and QA-specific probing tasks that reveal the information stored in each representation layer. Our qualitative analysis of hidden state visualizations provides additional insights into BERT's reasoning process. Our results show that the transformations within BERT go through phases that are related to traditional pipeline tasks. The system can therefore implicitly incorporate task-specific information into its token representations. Furthermore, our analysis reveals that fine-tuning has little impact on the models' semantic abilities and that prediction errors can be recognized in the vector representations of even early layers.

KEYWORDS

neural networks, transformers, explainability, word representation, natural language processing, question answering

ACM Reference format:
Anonymous Author(s). 2019. How Does BERT Answer Questions? A Layer-Wise Analysis of Transformer Representations. In *Proceedings of CIKM 2019: ACM International Conference on Information and Knowledge Management, Beijing, China., November 3rd-7th, 2019 (CIKM '19),* 10 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In recent months, Transformer models have become more and more prevalent in the field of Natural Language Processing. Originally they became popular for their improvements over RNNs in Machine Translation [37]. Now however, with the advent of large models and an equally large amount of pre-training being done, they have proven adept at solving many of the standard Natural Language Processing tasks. Main subject of this paper is BERT [9], arguably the most popular of the recent Transformer models and the first to display significant improvements over previous state-of-the-art models in a number of different benchmarks and tasks.

Problem of black box models. Deep Learning models achieve increasingly impressive results across a number of different domains, whereas their application to real-world tasks has been moving somewhat more slowly. One major impediment lies in the lack of transparency, reliability and prediction guarantees in these largely black box models.

While Transformers are commonly believed to be moderately interpretable through the inspection of their attention values, current research suggests that this may not always be the case [17]. This paper takes a different approach to the interpretation of said Transformer Networks. Instead of evaluating attention values, our approach examines the hidden states between encoder layers directly. There are multiple questions this paper will address:

(1) Do Transformers answer questions decompositionally, in a similar manner to humans?
(2) Do specific layers in a multi-layer Transformer network solve different tasks?
(3) What influence does fine-tuning have on a network's inner state?
(4) Can an evaluation of network layers help come to a conclusion on why and how a network failed to predict a correct answer?

We discuss these questions on the basis of fine-tuned models on standard QA datasets. We choose the task of Question Answering as an example of a complex downstream task that, as this paper will show, requires solving a multitude of other Natural Language Processing tasks. Additionally, it has been shown that other NLP tasks can be successfully framed as QA tasks [24], therefore our analysis should translate to these tasks as well. While this work focuses on the BERT architecture, we perform preliminary tests on the small GPT-2 model [30] as well, which yield similar results.

Contributions. With the goal to improve understanding of internal workings of Transformers we present the following contributions:

First, we propose a layer-wise visualisation of token representations that reveals information about the internal state of Transformer networks. This visualisation can be used to expose wrong predictions even in earlier layers or to show which parts of the context the model considered as Supporting Facts.

Second, we apply a set of general NLP Probing Tasks and extend them by the QA-specific tasks of Question Type Classification and Supporting Fact Extraction. This way we can analyse the abilities within BERT's layers and how they are impacted by fine-tuning.

Third, we show that BERT’s transformations go through similar phases, even if fine-tuned on different tasks. Information about general language properties is encoded in earlier layers of BERT and implicitly used to solve the downstream task at hand in later layers.

2 RELATED WORK

Transformer Models. Our analyses focus on BERT, which belongs to the group of Transformer networks, named after how representations are transformed throughout the network layers. We also partly include the more recent Transformer model GPT-2 [30]. This model represents OpenAI’s improved version of GPT [29] and while GPT-2 has not yet climbed leaderboards like BERT has, its larger versions have proven adept enough at the language modeling task, that Open-AI has decided not to release their pre-trained models. There are also other Transformer models of note, where a similar analysis might prove interesting in future work. Chief among them are the Universal Transformer [8] and TransformerXL [7], both of which aim to improve some of the flaws of the Transformer architecture by adding a recurrent inductive bias.

Interpretability and Probing. Explainability and Interpretability of neural models have become an increasingly large field of research. While there are a multitude of ways to approach these topics [10, 13, 21], we especially highlight relevant work in the area of research that builds and applies probing tasks and methodologies, post-hoc, to trained models. There have been a number of recent advances on this topic. While the majority of the current works aim to create or apply more general purpose probing tasks [3, 5, 34], BERT specifically has also been probed in previous papers. Tenney et al. [35] proposes a novel "edge-probing" framework consisting of nine different probing tasks and applies it to the contextualized word embeddings of ELMo, BERT and GPT-1. Both semantic and syntactic information is probed, but only pre-trained models are studied, and not specifically fine-tuned ones. A similar analysis [12] adds more probing tasks and addresses only the BERT architecture.

Qiao et al. [28] focus specifically on analysing BERT in the context of a Ranking task. The authors probe attention values in different layers and measure performance for representations build from different BERT layers. Like [35], they only discuss pre-trained models.

There has also been work which studies models not through probing tasks but through qualitative visual analysis. Zhang and Zhu [42] offer a survey of different approaches, though limited to CNNs. Nagamine et al. [26] explore phoneme recognition in DNNs by studying single node activations in the task of speech recognition. Hupkes et al. [16] go one step further, by not only doing a qualitative analysis, but also training diagnostic classifiers to support their hypotheses. Finally, Li et al. [19] take a look at word vectors and the importance of some of their specific dimensions on both sequence tagging and classification tasks.

The most closely related previous work is proposed by Liu et al. [22]. Here, the authors also perform a layer-wise analysis of BERT’s token representations. However, their work solely focuses on probing pre-trained models and disregards models fine-tuned on downstream tasks. Furthermore, it limits the analysis to the general transferability of the network and does not analyze the specific phases that BERT goes through.

Additionally, our work is motivated by Jain and Wallace [17]. In their paper, the authors argue that attention, at least in some cases, is not well suited to solve the issues of explainability and interpretability. They do so both by constructing adversarial examples and by a comparison with more traditional explainability methods. In supporting this claim, we propose revisiting evaluating hidden states and token representations instead.

3 BERT UNDER THE MICROSCOPE

We focus our analysis on fine-tuned BERT models. In order to understand which transformations the models apply to input tokens, we take two approaches: First, we analyse the transforming token vectors *qualitatively* by examining their positions in vector space. Second, we probe their language abilities on QA-related tasks to examine our results *quantitatively*.

3.1 Analysis of Transformed Tokens

The architecture of BERT and Transformer networks in general allows us to follow the transformations of each token throughout the network. We use this characteristic for an analysis of the changes that are being made to the tokens’ representations in every layer.

We use the following approach for a qualitative analysis of these transformations: We randomly select both correctly and falsely predicted samples from the test set of the respective dataset. For these samples we collect the hidden states from each layer while removing any padding. This results in the representation of each token throughout the model’s layers.

The model can transform the vector space freely throughout its layers and we do not have references for semantic meanings of positions within these vector spaces. Therefore we consider distances between token vectors as indication for semantic relations.

Dimensionality reduction. BERT’s pre-trained models use vector dimensions of 1024 (large model) and 512 (base model). In order to visualize relations between tokens, we apply dimensionality reduction and fit the vectors into two-dimensional space. To that end we apply T-distributed Stochastic Neighbor Embedding (t-SNE) [36], Principal Component Analysis (PCA) [11] and Independent Component Analysis (ICA) [4] to vectors in each layer. As the results of PCA reveal the most distinct clusters for our data, we use it to present our findings.

K-means clustering. In order to verify that clusters in 2D space represent the actual distribution in high-dimensional vector space, we additionally apply a k-means clustering [23]. We choose the number of clusters k in regard to the number of observed clusters in PCA, which vary over layers. The resulting clusters correspond with our observations in 2D space.

3.2 Probing BERT’s Layers

Our goal is to further understand the abilities of the model after each transformation. We therefore apply a set of semantic probing tasks to analyze which information is stored within the transformed tokens after each layer. We want to know whether specific layers

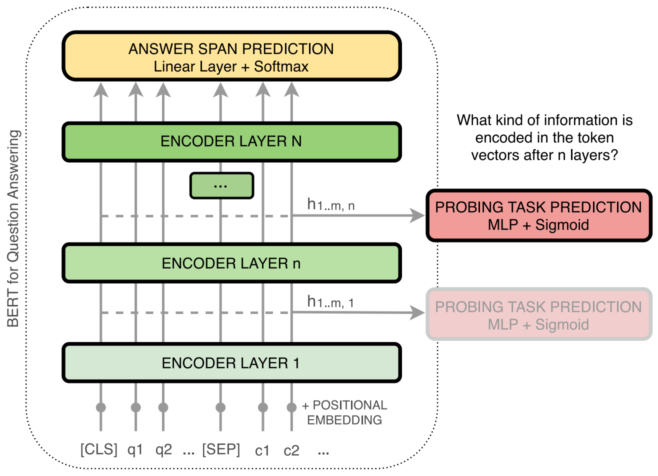


Figure 1: Schematic overview of the BERT architecture and our probing setup. Question and context tokens are processed by N encoder blocks with a Positional Embedding added before the first encoder. The output of the last layer is fed into a span prediction head consisting of a Linear layer and a Softmax. We use the hidden states of each layer as input to a set of probing tasks to examine the encoded information. Figure inspired by graphs from Alammar [2].

are *reserved* for specific tasks and how language information is maintained or forgotten by the model.

We use the principle of Edge Probing introduced by Tenney et al. [35]. Edge Probing translates core NLP tasks into classification tasks by focusing solely on their labeling part. This enables a standardized probing mechanism over a wide range of tasks. We adopt the tasks Named Entity Labeling, Coreference Resolution and Relation Classification from the original paper as they are prerequisites for language understanding and reasoning [40]. We add tasks of Question Type Classification and Supporting Fact Identification due to their importance for Question Answering in particular.[1]

Named Entity Labeling. Given a span of tokens the model has to predict the correct entity category. This is based on Named Entity Recognition but formulated as a Classification problem. The task was modeled by [35], annotations are based on the OntoNotes 5.0 corpus [39] and contain 18 entity categories.

Coreference Resolution. The Coreference task requires the model to predict whether two mentions within a text refer to the same entity. The task was built from the OntoNotes corpus and enhanced with negative samples by [35].

Relation Classification. In Relation Classification the model has to predict which relation type connects two known entities. The task was constructed by [35] with samples taken from the SemEval 2010 Task 8 dataset consisting of English web text and nine directional relation types.

[1]We will make the source code to all experiments publicly available.

Question Type Classification. A fundamental part of answering a question is to correctly identify its question type. For this Edge Probing task we use the Question Classification dataset constructed by Li and Roth [20] based on the TREC-10 QA dataset [38]. It includes 500 fine-grained types of questions within the larger groups of abbreviation, entity, description, human, location and numeric value. We use the whole question as input to the model with its question type as label.

Supporting Facts. The extraction of Supporting Facts is a main prerequisite for Question Answering tasks, especially in the multi-hop case. We examine what BERT's token transformations can tell us about the mechanism behind distinguishing distracting from important context parts.

To understand at which stage this distinction is done, we construct a probing task for identifying Supporting Facts. The model has to predict whether a sentence contains supporting facts regarding a specific question or whether it is irrelevant. Through this task we test the hypothesis that token representations contain information about their significance to the question.

Both HotpotQA and bAbI contain information about sentence-wise Supporting Facts for each question. SQuAD does not require multi-hop reasoning, we therefore consider the sentence containing the answer phrase the Supporting Fact. We also exclude all QA-pairs that only contain one context sentence. We construct a different probing task for each dataset in order to check their task-specific ability to recognize relevant parts. All samples are labeled sentence-wise with true if they are a supporting fact or false otherwise.

Probing Setup. Analogue to the authors of [35], we embed input tokens for each probing task sample with our fine-tuned BERT model. Contrary to previous work, we do this for all layers ($N = 12$ for BERT-base and $N = 24$ for BERT-large), using only the output embedding from n-th layer at step n. The concept of Edge Probing defines that only tokens of "labeled edges" (e.g. tokens of two related entities for Relation Classification) within a sample are considered for classification. These tokens are first pooled for a fixed-length representation and afterwards fed into a two-layer Multi-layer Perceptron (MLP) classifier, that predicts label-wise probability scores (e.g. for each type of relation). A schematic overview of this setting is shown in Figure 1. We perform the same steps on pre-trained BERT-base and BERT-large models without any fine-tuning. This enables us to identify which abilities the model learns during pre-training or fine-tuning.

4 DATASETS AND MODELS

4.1 Datasets

Our aim is to understand how BERT works on complex downstream tasks. Question Answering (QA) is one of such tasks that require a combination of multiple simpler tasks such as Coreference Resolution and Relation Modeling to arrive at the correct answer. We take three current Question Answering datasets into account, namely SQUAD [32], bAbI [40] and HotpotQA [41]. We intentionally choose three very different datasets to diversify the results of our analysis.

	SQuAD	bAbI
Question	What is a common punishment in the UK and Ireland?	What is Emily afraid of?
Answer	**detention**	**cats**
Context	**Currently detention is one of the most common punishments in schools in the United States, the UK, Ireland, Singapore and other countries.** It requires the pupil to remain in school at a given time in the school day (such as lunch, recess or after school); or even to attend school on a non-school day, e.g. "Saturday detention" held at some schools. During detention, students normally have to sit in a classroom and do work, write lines or a punishment essay, or sit quietly.	**Wolves are afraid of cats.** Sheep are afraid of wolves. Mice are afraid of sheep. Gertrude is a mouse. Jessica is a mouse. **Emily is a wolf.** Cats are afraid of sheep. Winona is a wolf.

Table 1: Samples from SQuAD dataset (left) and from Basic Deduction task (#15) of the bAbI dataset (right). Supporting Facts are printed in bold. The SQuAD sample can be solved by word matching and entity resolution, while the bAbI sample requires a logical reasoning step and cannot be solved by simple word matching. Figures in the further analysis will use these examples where applicable.

SQuAD. As one of the most popular QA tasks the SQuAD dataset contains 100.000 natural question-answer pairs on 500 Wikipedia articles. A new version of the dataset called SQuAD 2.0 [31] additionally includes unanswerable questions. We use the previous version SQuAD 1.1 for our experiments to concentrate on the base task of span prediction. In 2018 an ensemble of fine-tuned BERT models has bested the Human Baseline on this task. The dataset is characterised by questions that mainly require to resolve lexical and syntactic variations.

HotpotQA. This Multihop QA task contains 112.000 natural question-answer pairs. The questions are especially designed to combine information from multiple parts of a context. For our analysis we focus on the *distractor*-task of HotpotQA, in which the context is composed of both supporting and distracting facts with an average size of 900 words. As the pre-trained BERT model is restricted to an input size of 512 tokens, we reduce the amount of distracting facts by a factor of 2.7. We also leave out yes/no-questions (7% of questions) as they require additional specific architecture, diluting our analysis.

bAbI. The QA bAbI tasks are a set of artificial toy tasks developed to further understand the abilities of neural models. The 20 tasks require reasoning over multiple sentences (Multihop QA) and are modeled to include Positional Reasoning, Argument Relation Extraction and Coreference Resolution. The tasks strongly differ from the other QA tasks in their simplicity (e.g. vocabulary size of 230 and short contexts) and the artificial nature of sentences.

4.2 BERT and GPT-2

In this section we briefly discuss the models our analysis is based on, BERT [9] and GPT-2 [30]. Both of these models are Transformers that extend and improve on a number of different recent ideas. These include previous Transformer models [37][29], Semi-Supervised Sequence Learning [6], ELMo [27] and ULMFit [14]. Both have a similar architecture, and they each represent one half of the original encoder-decoder Transformer [37]. While GPT-2, like its predecessor, consists of only the decoder half, BERT uses a bidirectional variant of the original encoder. Each consists of a large number of Transformer blocks (12 for small GPT-2 and bert-base, 24 for bert-large), that in turn consist of a Self-Attention module, Feed Forward network, Layer Normalization and Dropout. On top of these encoder stacks we add a Sequence Classification head for the bAbI dataset and a Span Prediction head for the other datasets. Figure 1 depicts how these models integrate into our probing setup.

	SQuAD	HotpotQA Distr.	HotpotQA SP	bAbI
Baseline	77.2	66.0	66.0	42.0
BERT	87.9	56.8	80.4	93.4
GPT-2	74.9	54.0	64.6	99.9

Table 2: Results from fine-tuning BERT on QA tasks. Baselines are: BIDAF [33] for SQuAD, the LSTM Baseline for bAbI from [40] and the HotpotQA baseline from [41] for the two Hotpot tasks.

4.3 Applying BERT to Question Answering

We base our training code on the Pytorch implementation of BERT available at [15]. We use the publicly available pre-trained BERT models for our experiments. In particular, we study the monolingual models *bert-base-uncased* and *bert-large*. For GPT-2 the small model (117M Parameters) is used, as a larger model has not yet been released. However, we do not apply these models directly, and instead fine-tune them on each of our datasets.

Training Modalities. Regarding hyperparameters, we tune the learning rate, batch size and learning rate scheduling according to a grid search and train each model for 5 Epochs with evaluations on the development set every 1000 iterations. We then select the model of the best evaluation for further analysis. The input length chosen is 384 tokens for the bAbI and SQuAD tasks and the maximum of 512 tokens permitted by the pre-trained models' positional embedding for the HotpotQA tasks. For BAbI we evaluate both models that are trained on a single bAbI task and also a multitask model,

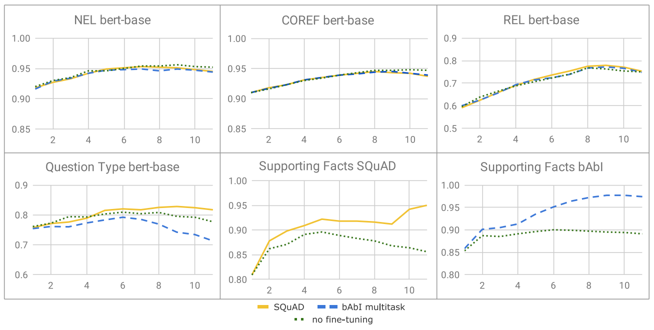


Figure 2: Probing Task results of BERT-base models in macro averaged F1 (Y-axis) over all layers (X-axis). Fine-tuning barely affects accuracy on NEL, COREF and REL indicating that those tasks are already sufficiently covered by pre-training. Performances on the Question Type task shows its relevancy for solving SQuAD, whereas it is not required for the bAbI tasks and the information is lost.

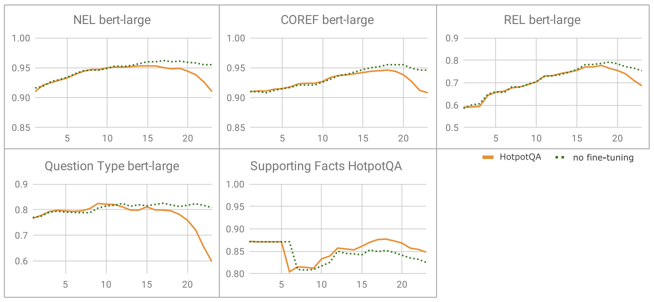


Figure 3: Probing Task results of BERT-large models in macro averaged F1 (Y-axis) over all layers (X-axis). Performance of HotpotQA model is mostly equal to the model without fine-tuning, but information is dropped in last layers in order to fit the Answer Selection task.

that was trained on the data of all 20 tasks. We further distinguish between two settings: Span prediction, which we include for better comparison with the other datasets, and Sequence Classification, which is the more common approach to bAbI. In order to make span prediction work, we append all possible answers to the end of the base context, since not all answers can be found in the context by default. For HotpotQA, we also distinguish between two tasks. In the *HotpotQA Support Only* (SP) task, we use only the sentences labeled as Supporting Facts as the question context. This simplifies the task, but more importantly it reduces context length and increases our ability to distinguish token vectors. Our *HotpotQA Distractor* task is closer to the original HotpotQA task. It includes distracting sentences in the context, but only enough to not exceed the 512 token limit.

5 RESULTS AND DISCUSSION

Training Results. Table 2 shows the evaluation results of our best models. Accuracy on the SQuAD task is close to human performance, indicating that the model can fulfill all sub-tasks required to answer SQuAD's questions. As expected the tasks derived from HotpotQA prove much more challenging, with the distractor setting being the most difficult to solve. Unsurprisingly too, bAbI was easily solved by both BERT and GPT-2. While GPT-2 performs significantly worse in the more difficult tasks of SQuAD and HotpotQA, it does considerably better on bAbi reducing the validation error to nearly 0. Most of BERT's error in the bAbI multi-task setting comes from tasks 17 and 19. Both of these tasks require positional or geometric reasoning, thus it is reasonable to assume that this is a skill where GPT-2 improves on BERT's reasoning capabilities.

Presentation of Analysis Results. The qualitative analysis of vector transformations reveals a range of recurring patterns. In the following, we present these patterns by two representative samples from the SQuAD and bAbI task dataset described in Table 1. Examples from HotpotQA can be found in the supplementary material as they require more space due to the larger context.

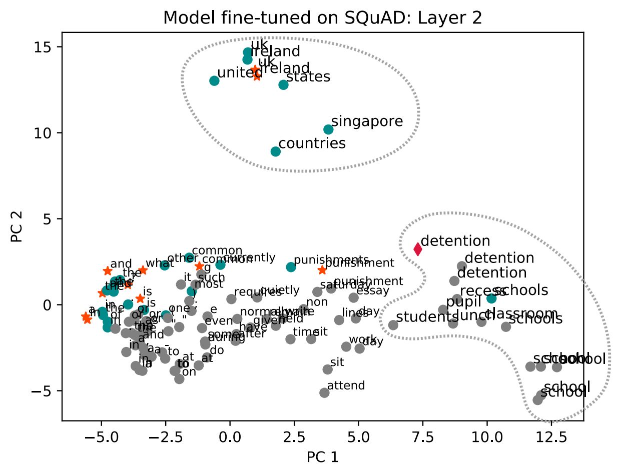
Results from probing tasks are displayed in Figures 2 and 3. We compare results in macro-averaged F1 over all network layers. Figure 2 shows results from three models of BERT-base with twelve layers: Fine-tuned on SQuAD,on bAbI tasks and without fine-tuning. Figure 3 reports results of two models based on BERT-large with 24 layers: Fine-tuned on HotpotQA and without fine-tuning.

5.1 Phases of BERT's Transformations

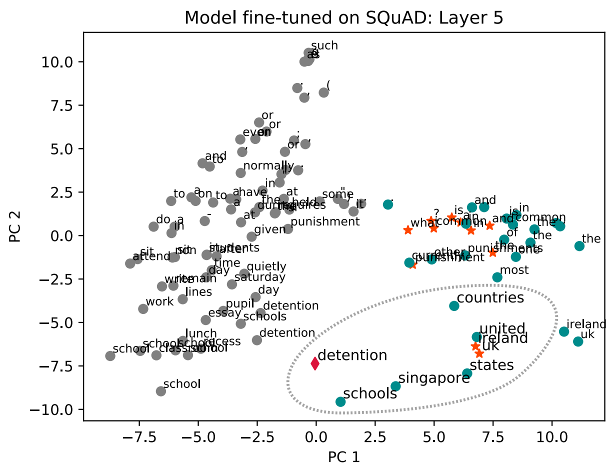
The PCA representations of tokens in different layers suggest that the model is going through multiple phases while answering a question. We observe these phases in all three selected QA tasks despite their diversity. These findings are supported by results of the applied probing tasks. We present the four phases in the following paragraphs and describe how our experimental results are linked.

(1) Semantic Clustering. Early layers within the BERT-based models group tokens into topical clusters. Figures 4a and 5a reveal this behaviour and show the second layer of each model. Resulting vector spaces are similar in nature to embedding spaces from e.g. Word2Vec [25] and hold little task-specific information. Therefore, these initial layers reach low accuracy on semantic probing tasks, as shown in Figures 2 and 3. BERT's early layers can be seen as an implicit replacement of embedding layers common in neural network architectures.

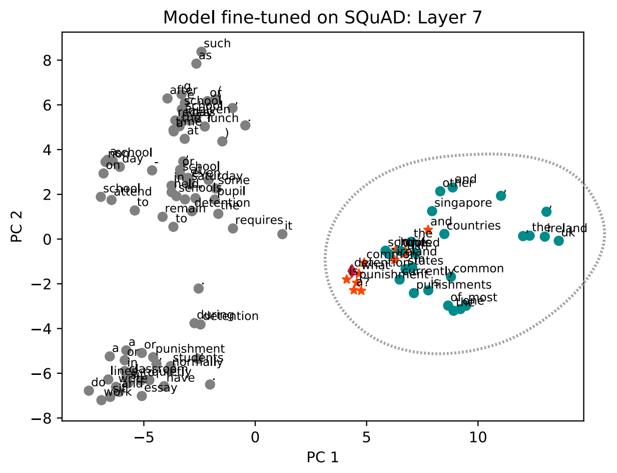
(2) Connecting Entities with Mentions and Attributes. In the middle layers of the observed neural networks, we see clusters of entities that are less connected by their topical similarity. Rather, they are connected by their relation within a certain input context. These task-specific clusters appear to already include a filtering of question-relevant entities. Figure 4b shows a cluster with words like *countries, schools, detention* and country names, in which 'detention' is a common practice in schools. This cluster helps to solve the question *"What is a common punishment in the UK and Ireland?"*. Another question-related cluster is shown in Figure 5b. The main challenge within this sample is to identify the two facts that *Emily is a wolf* and *Wolves are afraid of cats.* The highlighted cluster implies that *Emily* has been recognized as a relevant entity that holds a relation to the entity *Wolf.* The cluster also contains other



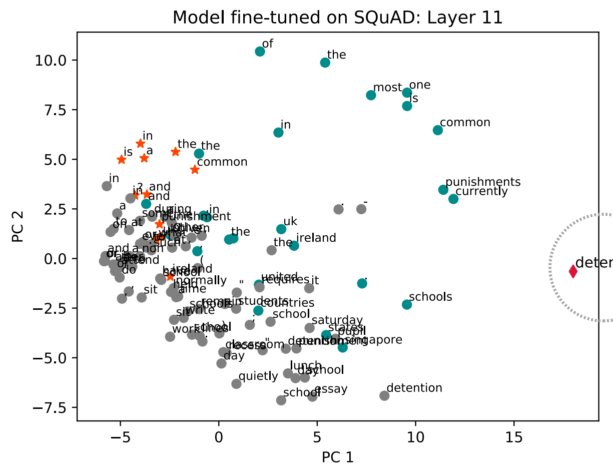
(a) SQuAD Phase 1: Semantic Clustering. We observe a topical cluster with 'school'-related and another with 'country'-related tokens.



(b) SQuAD Phase 2: Entity Matching. The marked cluster contains matched tokens 'detention', 'schools' and the countries that are applying this practice.



(c) SQuAD Phase 3: Question-Fact Matching. The question tokens form a cluster with the Supporting Fact tokens.



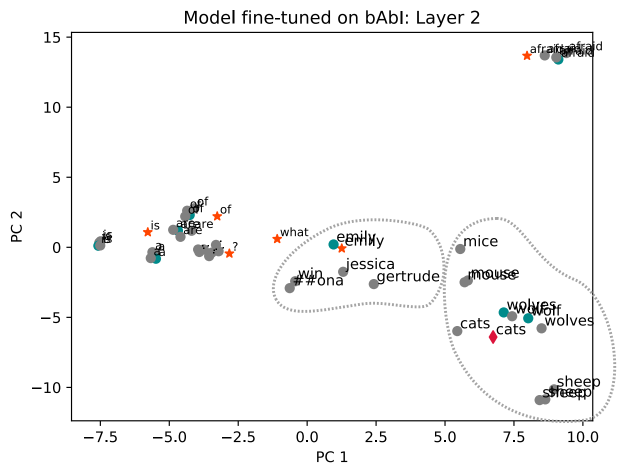
(d) SQuAD Phase 4: Answer Extraction. The answer token 'detention' is separated from other tokens.

Figure 4: BERT's Transformation Phases for the SQuAD example from Table 1. Answer token: Red diamond-shaped. Question Tokens: Orange star-shaped. Supporting Fact tokens: Dark Cyan. Prominent clusters are circled. The model passes through different phases in order to find the answer token, which is extracted in the last layer (#11).

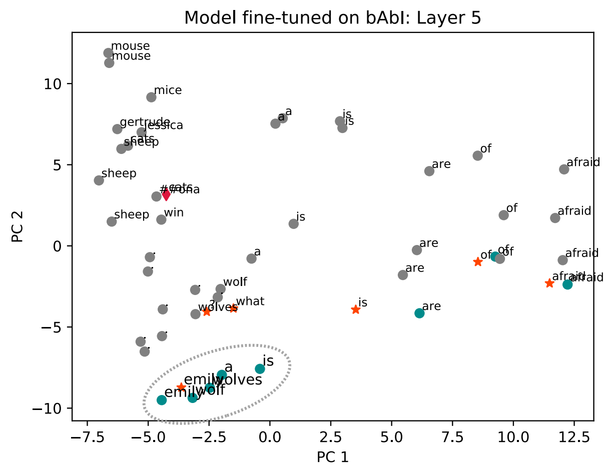
mentions of these entities including the plural form *Wolves.* We observe similar clusters in the HotpotQA model, which includes more cases of coreferences.

The probing results support these observations. The model's ability to recognize entities (Named Entity Labeling), to identify their mentions (Coreference Resolution) and to find relations (Relation Recognition) improves until higher network layers. Figure 6 visualizes these abilities. Information about Named Entities is learned first, whereas recognizing coreferences or relations are more difficult tasks and require input from additional layers until the model's performance peaks. These patterns are equally observed in the results from BERT-base models and BERT-large models.

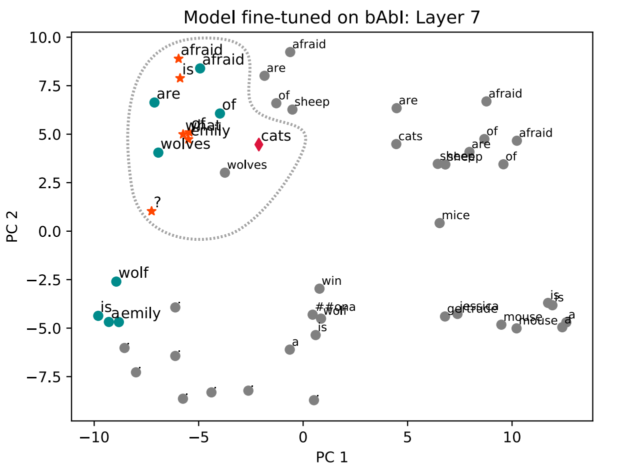
(3) Matching Questions with Supporting Facts. Identifying relevant parts of the context is crucial for Question Answering and Information Retrieval in general. In traditional pipeline models this step is often achieved by filtering context parts based on their similarity to the question [18]. We observe that BERT models perform a corresponding step by transforming the tokens so that question tokens are matched onto relevant context tokens. Figures 4c and 5c show two examples in which the model transforms the token representation of question and Supporting Facts into the same area of the vector space. Some samples show this behaviour in lower layers. However, results from our probing tasks reveal that the models hold the strongest ability to distinguish relevant from irrelevant information wrt. the question in their higher layers. Figure 2



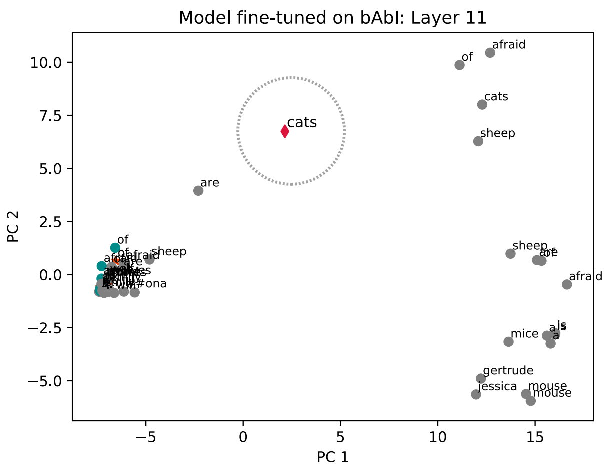
(a) bAbI Phase 1: Semantic Clustering. Names and animals are clustered.



(b) bAbI Phase 2: Entity Matching. The determining relation between the entities ’Emily’ and ’Wolf’ is resolved in a cluster.



(c) bAbI Phase 3: Question-Fact Matching. In this case the question tokens match with a subset of Supporting Facts (’Wolves are afraid of cats’). The subset is decisive of the answer.



(d) bAbI Phase 4: Answer Extraction. The answer token ’cats’ is separated from other tokens.

Figure 5: BERT’s Transformation Phases for the bAbI example from Table 1. The phases are equal to what we observe in SQuAD and HotpotQA samples: The formed clusters in the first layers show general language abilities, while the last layers are more task-specific.

demonstrates how the performance for this task increases over successive layers for SQuAD and bAbI. Performance of the fine-tuned HotpotQA model in Figure 3 is less distinct from the model without fine-tuning and does not reach high accuracy.[2] This inability indicates why the BERT model does not perform well on this dataset as it is not able to identify the correct Supporting Facts.

The vector representations enable us to tell which facts a model considered important (and therefore matched with the question). This helps retracing decisions and makes the model more transparent.

[2]Note that the model only predicts the majority class in the first five layers and thereby reaches a decent accuracy without really solving the task.

(4) Answer Extraction. In the last network layers we see that the model dissolves most of the previous clusters. Here, the model separates the correct answer tokens, and sometimes other possible candidates, from the rest of the tokens. The remaining tokens form one or multiple homogeneous clusters. The vector representation at this point is largely task-specific and learned during fine-tuning. This becomes visible through the performance drop in general NLP probing tasks, visualized in Figure 6. We especially observe this loss of information in last-layer representations in the large BERT-model fine-tuned on HotpotQA, as shown in Figure 3. While the model without fine-tuning still performs well on tasks like NEL or COREF, the fine-tuned model loses this ability.

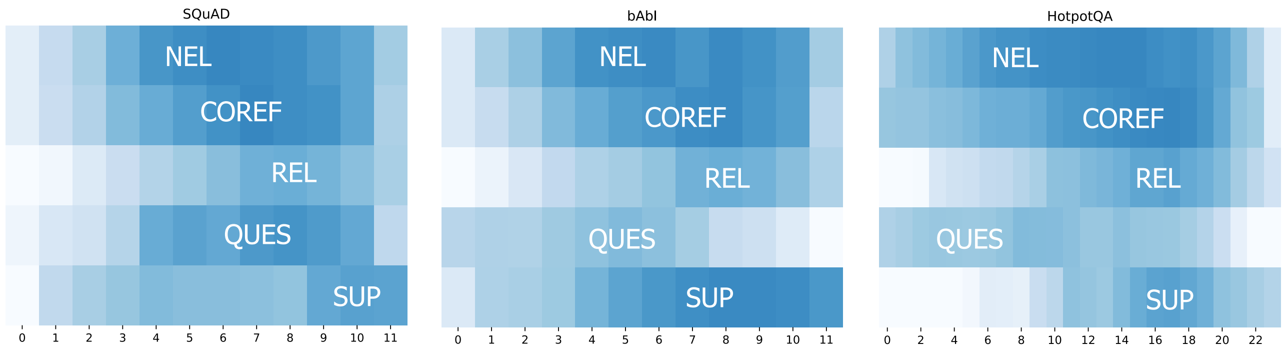


Figure 6: Phases of BERT's language abilities. Higher saturation denotes higher accuracy on probing tasks. Values are normalized over tasks on the Y-axis. X-axis depicts layers of BERT. NEL: Named Entity Labeling, COREF: Coreference Resolution, REL: Relation Classification, QUES: Question Type Classification; SUP: Supporting Fact Extraction. All three tasks exhibit the same patterns, except from QUES, which is solved earlier by the HotpotQA model based on BERT-large. NEL is solved first, while performance on COREF and REL reaches its peak in later layers. Distinction of important facts (SUP) happens within the last layers.

Analogies to Human Reasoning. The phases of answering questions can be compared to the human reasoning process, including decomposition of input into parts [1]. The first phase of semantic clustering represents our basic knowledge of language and the second phase how a human reader builds relations between parts of the context to connect information needed for answering a question. Separation of important from irrelevant information (phase 3) and grouping of potential answer candidates (phase 4) are also known from human reasoning. However, the order of these steps might differ from the human abstraction. One major difference is that while humans read sequentially, BERT can see all parts of the input at once. Thereby it is able to run multiple processes and phases concurrently depending on the task at hand. Figure 6 shows how the tasks overlap during the answering process.

5.2 Comparison to GPT-2

In this section we compare our insights from the BERT models to the GPT-2 model. We focus on the qualitative analysis of token representations and leave the application of probing tasks for future work. One major difference between GPT-2's and BERT's hidden states is that GPT-2 seems to give particular attention to the first token of a sequence. While in our QA setup this is often the question word, this also happens in cases where it is not. During dimensionality reduction this results in a separation of two clusters, namely the first token and all the rest. This problem holds true for all layers of GPT-2 except for the Embedding Layer, the first Transformer block and the last one. For this reason we mask the first token during dimensionality reduction in further analysis.

Figure 7 shows an example of the last layer's hidden state for our bAbI example. Like BERT, GPT-2 also separates the relevant Supporting Facts and the question in the vector space. Additionally, GPT-2 extracts another sentence, which is not a Supporting Fact, but is similar in meaning and semantics. In contrast to BERT, the correct answer "cats" is not particularly separated and instead simply left as part of its sentence. These findings in GPT-2 suggest that our analysis extends beyond the BERT architecture and hold true for other Transformer networks as well. Our future work will include more probing tasks to confirm this initial observation.

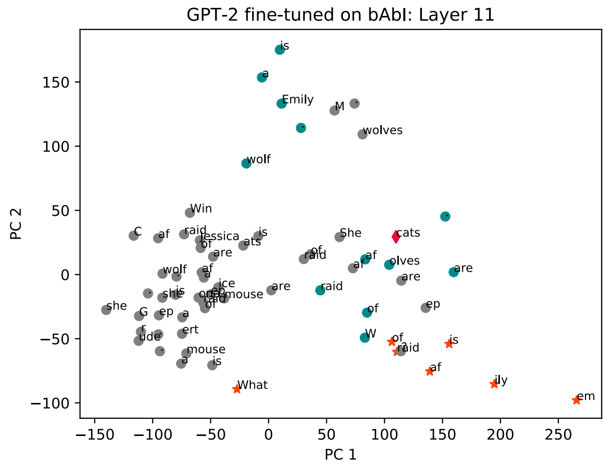


Figure 7: bAbI Example of the Answer Extraction phase in GPT-2. Both the question and Supporting Fact are extracted, but the correct answer is not fully separated as in BERT's last layers. Also a potential candidate Supporting Fact in "Sheep are afraid of Wolves" is separated as well.

5.3 Additional Findings

Observation of Failure States. One important aspect of explainable Neural Networks is to answer the questions of when, why, and how the network fails. Our visualizations are not only able to show such failure states, but even the rough difficulty of a specific task can be discerned by a glance at the hidden state representations.

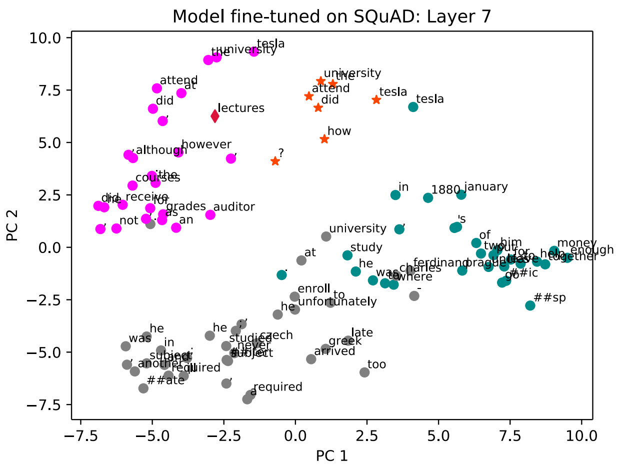


Figure 8: BERT SQuAD example of a falsely selected answer based on the matching of the wrong Supporting Fact. The predicted answer 'lectures' is matched onto the question as a part of this incorrect fact (magenta), while the actual Supporting Fact (cyan) is not particularly separated.

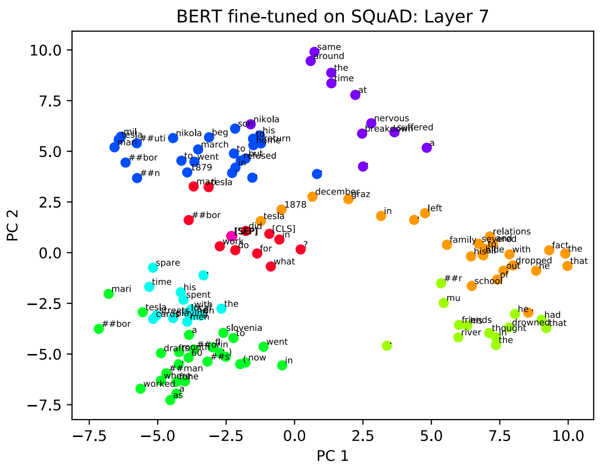


Figure 9: BERT SQuAD example Layer 7. Tokens are color-coded by sentence. This visualization shows that tokens are clustered by their original sentence membership suggesting far reaching importance of the positional embedding.

While for correct predictions the transformations run through the phases discussed in previous sections, for wrong predictions there exist two possibilities: If a candidate answer was found that the network has a reasonable amount of confidence in, the phases will look very similar to a correct prediction, but now centering on the wrong answer. Inspecting early layers in this case can give insights towards the reason why the wrong candidate was chosen, e.g. wrong Supporting Fact selected, misresolution of coreferences etc. An example of this is shown in Figure 8, where a wrong answer is based on the fact that the wrong Supporting Fact was matched with the question in early layers.

If network confidence is low however, which is often the case when the predicted answer is far from the actual answer, the transformations do not go through the phases discussed earlier. The vector space is still transformed in each layer, but tokens are mostly kept in a single homogenous cluster. If something *is* extracted it usually has little to do with the prediction. In some cases, especially when the confidence of the network is low in general, the network maintains Phase (1), 'Semantic Clustering' analogue to Word2Vec, even in later layers. An example is depicted in the supplementary material.

Impact of Fine-tuning. Figures 2 and 3 show how little impact fine-tuning has on the core NLP abilities of the model. The pre-trained model already holds sufficient information about words and their relations, which is the reason it works well in multiple downstream tasks. Fine-tuning only applies small weight changes and forces the model to forget some information in order to fit specific tasks. However, the model does not forget much of the previously learned encoding when fitting the QA task, which indicates why the Transfer Learning approach proves successful.

Maintained Positional Embedding. It is well known that the positional embedding is a very important factor in the performance of Transformer networks. It solves one major problem that Transformers have in comparison with RNNs, that they lack sequential information [37]. Our visualizations support this importance and show that even though the positional embedding is only added once before the first layer, its effects are maintained even into very late layers depending on the task. Figure 9 demonstrates this behavior on the SQuAD dataset.

Abilities to resolve Question Type. The performance curves regarding the Question Type probing task illustrate another interesting result. Figure 2 demonstrates that the model fine-tuned on SQuAD outperforms the base model from layer 5 onwards. This indicates the relevancy of resolving the question type for the SQuAD task, which leads to an improved ability after fine-tuning. The opposite is the case for the model fine-tuned on the bAbI tasks, which loses part of its ability to distinguish question types during fine-tuning. This is likely caused by the static structure of bAbI samples, in which the answer candidates can be recognized by sentence structure and occurring word patterns rather than by the question type. Surprisingly, we see that the model fine-tuned on HotpotQA does not outperform the model without fine-tuning in Figure 3. Both models can solve the task in earlier layers, which suggests that the ability to recognize question types is pre-trained in BERT-large.

6 CONCLUSION AND FUTURE WORK

Our work reveals important findings about the inner functioning of Transformer networks. The impact of these findings and how future work can build upon them is described in the following:

Interpretability. The qualitative analysis of token vectors reveals that there is indeed interpretable information stored within the hidden states of Transformer models. This information can be used to identify misclassified examples and model weaknesses. It also provides clues about which parts of the context the model considered important for answering a question - a crucial part of decision legitimisation. We leave the development of methods to further process this information for future work.

Transferability. We further show that lower layers might be more applicable to certain problems than later ones. For a Transfer Learning task, this means layer depth should be chosen individually depending on the task at hand. We also suggest further work regarding skip connections in Transformer layers to examine whether direct information transfer between non-adjacent layers (that solve different tasks) can be of advantage.

Modularity. Our findings support the hypothesis that not only do different phases exist in Transformer networks, but that specific layers seem to solve different problems. This hints at a kind of modularity that can potentially be exploited in the training process. For example, it could be beneficial to fit parts of the network to specific tasks in pre-training, instead of using an end-to-end language model task.

Our work aims towards revealing some of the internal processes within Transformer-based models. We suggest to direct further research at thoroughly understanding state-of-the-art models and the way they solve downstream tasks, in order to improve on them.

REFERENCES

[1] Lotfi A Zadeh. 1997. Zadeh, L.A.: Toward a Theory of Fuzzy Information Granulation and Its Centrality in Human Reasoning and Fuzzy Logic. Fuzzy Sets and Systems. *ELSEVIER Fuzzy Sets and Systems* 90 (1997).

[2] Jay Alammar. 2018. The Illustrated Transformer [Blog post]. (2018). https://jalammar.github.io/illustrated-transformer/

[3] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James R. Glass. 2017. What do Neural Machine Translation Models Learn about Morphology?. In *Proceedings of ACL 2017*.

[4] Pierre Comon. 1994. Independent component analysis, A new concept? *Signal Processing* 36 (1994).

[5] Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of LREC 2018*.

[6] Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised Sequence Learning. In *Proceedings of NIPS 2015*.

[7] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *CoRR* (2019).

[8] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2018. Universal Transformers. In *Proceedings of SMACD 2018*.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* (2018).

[10] F. K. Došilović, M. Brčić, and N. Hlupić. 2018. Explainable artificial intelligence: A survey. In *MIPRO 2018*.

[11] Karl Pearson F.R.S. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (1901).

[12] Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. *CoRR* (2019).

[13] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. 2018. A Survey Of Methods For Explaining Black Box Models. *ACM Comput. Surv.* (2018).

[14] Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned Language Models for Text Classification. *CoRR* (2018).

[15] Huggingface. 2018. pytorch-pretrained-BERT. (2018). https://github.com/huggingface/pytorch-pretrained-BERT

[16] Dieuwke Hupkes, Sara Veldhoen, and Willem H. Zuidema. 2017. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. In *Proceedings of IJCAI 2018*.

[17] Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of NAACL 2019*.

[18] Dan Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Chapter 23*. Prentice Hall series in artificial intelligence, Vol. 2. Prentice Hall, Pearson Education International.

[19] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding Neural Networks through Representation Erasure. *CoRR* (2016).

[20] Xin Li and Dan Roth. 2002. Learning Question Classifiers. In *Proceedings of COLING 2002*.

[21] Zachary Chase Lipton. 2016. The Mythos of Model Interpretability. *ACM Queue* (2016).

[22] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A. Smith. 2019. Linguistic Knowledge and Transferability of Contextual Representations. In *Proceedings of NAACL 2019*.

[23] Stuart P. Lloyd. 1982. Least squares quantization in PCM. *IEEE Trans. Information Theory* (1982).

[24] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The Natural Language Decathlon: Multitask Learning as Question Answering. *CoRR* (2018).

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Workshop Track Proceedings of ICLR 2013*.

[26] Tasha Nagamine, Michael Seltzer, and Nima Mesgarani. 2015. Exploring How Deep Neural Networks Form Phonemic Categories. In *Proceedings of INTERSPEECH 2015*.

[27] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT 2018*.

[28] Yifan Qiao, Chenyan Xiong, Zheng-Hao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. *CoRR* (2019).

[29] Alec Radford. 2018. Improving Language Understanding by Generative Pre-Training. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[30] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019). https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[31] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of ACL 2018*.

[32] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *Proceedings of EMNLP 2016*.

[33] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. [n. d.]. Bidirectional Attention Flow for Machine Comprehension. In *Proceedings of ICLR 2017*.

[34] Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does String-Based Neural MT Learn Source Syntax?. In *Proceedings of EMNLP 2016*.

[35] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *Proceedings of ICLR 2019*.

[36] Laurens van der Maaten. 2009. Learning a Parametric Embedding by Preserving Local Structure. In *Proceedings of AISTATS 2009*.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of NIPS 2017*.

[38] Ellen Voorhees. 2001. Overview of TREC 2001. In *Proceedings of TREC 2001*.

[39] Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. *OntoNotes: A Large Training Corpus for Enhanced Processing*. Springer, Heidelberg.

[40] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2016. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *Proceedings of ICLR 2016*.

[41] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of EMNLP 2018*.

[42] Quan-shi Zhang and Song-chun Zhu. 2018. Visual interpretability for deep learning: a survey. *Frontiers of IT & EE* (2018).