What Does BERT Look At? An Analysis of BERT's Attention

Kevin Clark[†] **Urvashi Khandelwal**[†] **Omer Levy**[‡] **Christopher D. Manning**[†]
[†]Computer Science Department, Stanford University
[‡]Facebook AI Research
{kevclark,urvashik,manning}@cs.stanford.edu
omerlevy@fb.com

Abstract

Large pre-trained neural networks such as BERT have had great recent success in NLP, motivating a growing body of research investigating what aspects of language they are able to learn from unlabeled data. Most recent analysis has focused on model outputs (e.g., language model surprisal) or internal vector representations (e.g., probing classifiers). Complementary to these works, we propose methods for analyzing the attention mechanisms of pre-trained models and apply them to BERT. BERT's attention heads exhibit patterns such as attending to delimiter tokens, specific positional offsets, or broadly attending over the whole sentence, with heads in the same layer often exhibiting similar behaviors. We further show that certain attention heads correspond well to linguistic notions of syntax and coreference. For example, we find heads that attend to the direct objects of verbs, determiners of nouns, objects of prepositions, and coreferent mentions with remarkably high accuracy. Lastly, we propose an attention-based probing classifier and use it to further demonstrate that substantial syntactic information is captured in BERT's attention.

1 Introduction

Large pre-trained language models achieve very high accuracy when fine-tuned on supervised tasks (Dai and Le, 2015; Peters et al., 2018; Radford et al., 2018), but it is not fully understood why. The strong results suggest pre-training teaches the models about the structure of language, but what specific linguistic features do they learn?

Recent work has investigated this question by examining the *outputs* of language models on carefully chosen input sentences (Linzen et al., 2016) or examining the internal *vector representations* of the model through methods such as probing classifiers (Adi et al., 2017; Belinkov et al., 2017). Complementary to these approaches, we study[1] the *attention maps* of a pre-trained model. Attention (Bahdanau et al., 2015) has been a highly successful neural network component. It is naturally interpretable because an attention weight has a clear meaning: how much a particular word will be weighted when computing the next representation for the current word. Our analysis focuses on the 144 attention heads in BERT[2] (Devlin et al., 2019), a large pre-trained Transformer (Vaswani et al., 2017) network that has demonstrated excellent performance on many tasks.

We first explore generally how BERT's attention heads behave. We find that there are common patterns in their behavior, such as attending to fixed positional offsets or attending broadly over the whole sentence. A surprisingly large amount of BERT's attention focuses on the deliminator token [SEP], which we argue is used by the model as a sort of no-op. Generally we find that attention heads in the same layer tend to behave similarly.

We next probe each attention head for linguistic phenomena. In particular, we treat each head as a simple no-training-required classifier that, given a word as input, outputs the most-attended-to other word. We then evaluate the ability of the heads to classify various syntactic relations. While no single head performs well at many relations, we find that particular heads correspond remarkably well to particular relations. For example, we find heads that find direct objects of verbs, determiners of nouns, objects of prepositions, and objects of possessive pronouns with $>75\%$ accuracy. We perform a similar analysis for coreference resolution, also finding a BERT head that performs quite well. These results are intriguing because the behavior of the attention heads emerges purely from self-supervised training on unlabeled data, without explicit supervision for syntax or coreference.

[1]Code will be released at https://github.com/clarkkev/attention-analysis.

[2]We use the English base-sized model.

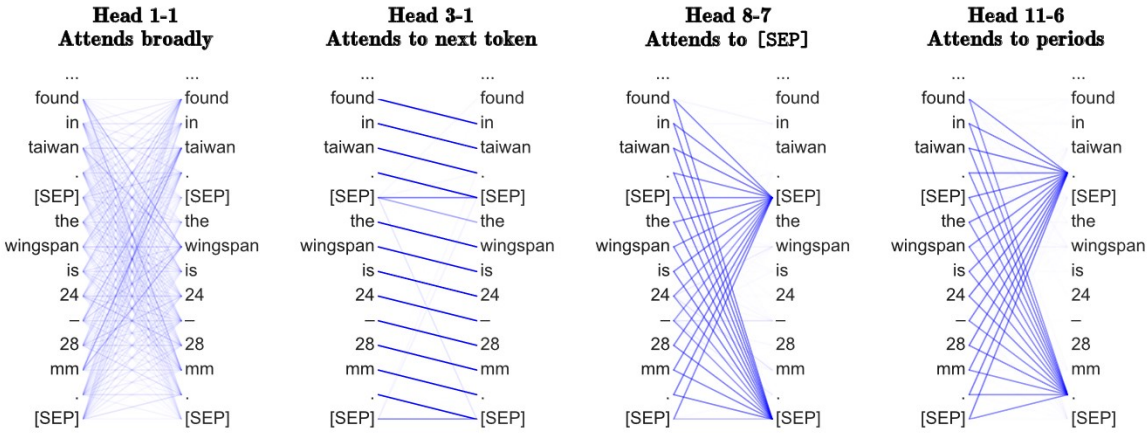

Figure 1: Examples of heads exhibiting the patterns discussed in Section 3. The darkness of a line indicates the strength of the attention weight (some attention weights are so low they are invisible).

Our findings show that particular heads specialize to specific aspects of syntax. To get a more overall measure of the attention heads' syntactic ability, we propose an attention-based probing classifier that takes attention maps as input. The classifier achieves 77 UAS at dependency parsing, showing BERT's attention captures a substantial amount about syntax. Several recent works have proposed incorporating syntactic information to improve attention (Eriguchi et al., 2016; Chen et al., 2018; Strubell et al., 2018). Our work suggests that to an extent this kind of syntax-aware attention already exists in BERT, which may be one of the reason for its success.

2 Background: Transformers and BERT

Although our analysis methods are applicable to any model that uses an attention mechanism, in this paper we analyze BERT (Devlin et al., 2019), a large Transformer (Vaswani et al., 2017) network. Transformers consist of multiple layers where each layer contains multiple attention heads. An attention head takes as input a sequence of vectors $h = [h_1, ..., h_n]$ corresponding to the n tokens of the input sentence. Each vector h_i is transformed into query, key, and value vectors q_i, k_i, v_i through separate linear transformations. The head computes attention weights α between all pairs of words as softmax-normalized dot products between the query and key vectors. The output o of the attention head is a weighted sum of the value vectors.

$$\alpha_{ij} = \frac{\exp\left(q_i^T k_j\right)}{\sum_{l=1}^{n} \exp\left(q_i^T k_l\right)} \qquad o_i = \sum_{j=1}^{n} \alpha_{ij} v_j$$

Attention weights can be viewed as governing how "important" every other token is when producing the next representation for the current token.

BERT is pre-trained on 3.3 billion tokens of English text to perform two tasks. In the "masked language modeling" task, the model predicts the identities of words that have been masked-out of the input text. In the "next sentence prediction" task, the model predicts whether the second half of the input follows the first half of the input in the corpus, or is a random paragraph. Further training the model on supervised data results in impressive performance across a variety of tasks ranging from sentiment analysis to question answering. An important detail of BERT is the preprocessing used for the input text. A special token [CLS] is added to the beginning of the text and another token [SEP] is added to the end. If the input consists of multiple separate texts (e.g., a reading comprehension example consists of a separate question and context), [SEP] tokens are also used to separate them. As we show in the next section, these special tokens play an important role in BERT's attention. We use the "base" sized BERT model, which has 12 layers containing 12 attention heads each. We use <layer>-<head_number> to denote a particular attention head.

3 Surface-Level Patterns in Attention

Before looking at specific linguistic phenomena, we first perform an analysis of surface-level patterns in how BERT's attention heads behave. Examples of heads exhibiting these patterns are shown in Figure 1.

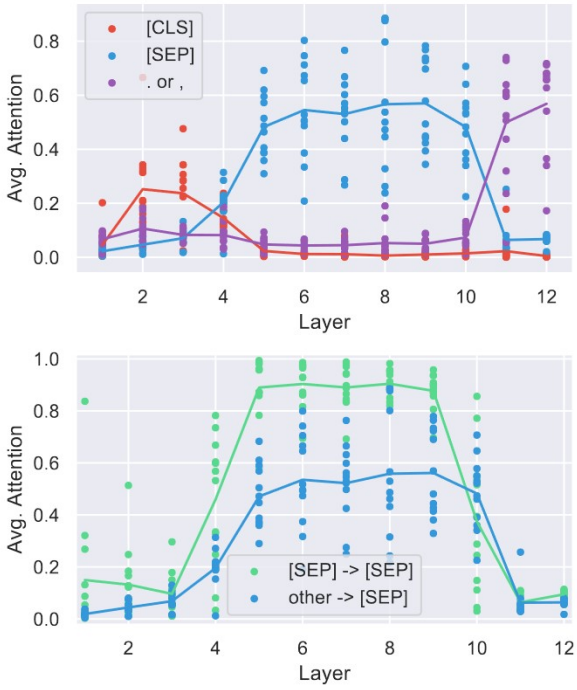

Figure 2: Each point corresponds to the average attention a particular BERT attention head puts toward a token type. Above: heads often attend to "special" tokens. Early heads attend to [CLS], middle heads attend to [SEP], and deep heads attend to periods and commas. Often more than half of a head's total attention is to these tokens. Below: heads attend to [SEP] tokens even more when the current token is [SEP] itself.

Setup. We extract the attention maps from BERT-base over 1000 random Wikipedia segments. We follow the setup used for pre-training BERT where each segment consists of at most 128 tokens corresponding to two consecutive paragraphs of Wikipedia (although we do not mask out input words or as in BERT's training). The input presented to the model is [CLS]<paragraph-1>[SEP]<paragraph-2>[SEP].

3.1 Relative Position

First, we compute how often BERT's attention heads attend to the current token, the previous token, or the next token. We find that most heads put little attention on the current token. However, there are heads that specialize to attending heavily on the next or previous token, especially in earlier layers of the network. In particular four attention heads (in layers 2, 4, 7, and 8) on average put >50% of their attention on the previous token and five attention heads (in layers 1, 2, 2, 3, and 6) put >50% of their attention on the next token.

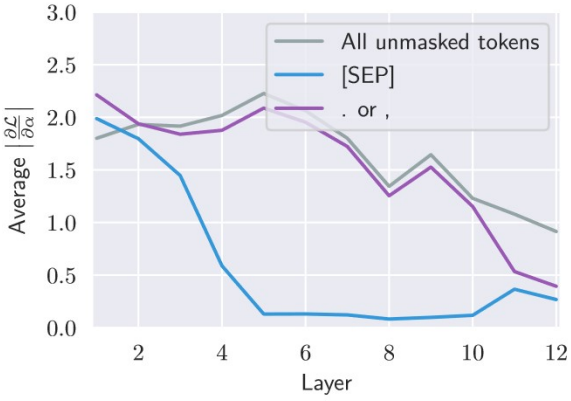

Figure 3: Gradient-based feature importance estimates for attention to [SEP], periods/commas, and other tokens.

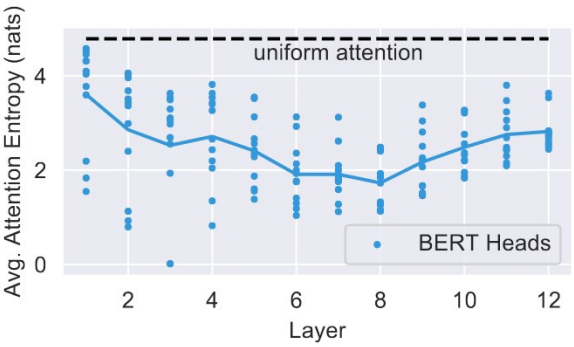

Figure 4: Entropies of attention distributions. In the first layer there are particularly high-entropy heads that produce bag-of-vector-like representations.

3.2 Attending to Separator Tokens

Interestingly, we found that a substantial amount of BERT's attention focuses on a few tokens (see Figure 2). For example, over half of BERT's attention in layers 6-10 focuses on [SEP]. To put this in context, since most of our segments are 128 tokens long, the average attention for a token occurring twice in a segments like [SEP] would normally be around 1/64. [SEP] and [CLS] are guaranteed to be present and are never masked out, while periods and commas are the most common tokens in the data excluding "the," which might be why the model treats these tokens differently. A similar pattern occurs for the uncased BERT model, suggesting there is a systematic reason for the attention to special tokens rather than it being an artifact of stochastic training.

One possible explanation is that [SEP] is used to aggregate segment-level information which can then be read by other heads. However, further analysis makes us doubtful this is the case. If this

explanation were true, we would expect attention heads processing [SEP] to attend broadly over the whole segment to build up these representations. However, they instead almost entirely (more than 90%; see bottom of Figure 2) attend to themselves and the other [SEP] token. Furthermore, qualitative analysis (see Figure 5) shows that heads with specific functions attend to [SEP] when the function is not called for. For example, in head 8-10 direct objects attend to their verbs. For this head, non-nouns mostly attend to [SEP]. Therefore, we speculate that attention over these special tokens might be used as a sort of "no-op" when the attention head's function is not applicable.

To further investigate this hypothesis, we apply gradient-based measures of feature importance (Sundararajan et al., 2017). In particular, we compute the magnitude of the gradient of the loss from BERT's masked language modeling task with respect to each attention weight. Intuitively, this value measures how much changing the attention to a token will change BERT's outputs. Results are shown in Figure 3. Starting in layer 5 – the same layer where attention to [SEP] becomes high – the gradients for attention to [SEP] become very small. This indicates that attending more or less to [SEP] does not substantially change BERT's outputs, supporting the theory that attention to [SEP] is used as a no-op for attention heads.

3.3 Focused vs Broad Attention

Lastly, we measure whether attention heads focus on a few words or attend broadly over many words. To do this, we compute the average entropy of each head's attention distribution (see Figure 4). We find that some attention heads, especially in lower layers, have very broad attention. These high-entropy attention heads typically spend at most 10% of their attention mass on any single word. The output of these heads is roughly a bag-of-vectors representation of the sentence.

We also measured entropies for all attention heads from only the [CLS] token. While the average entropies from [CLS] for most layers are very close to the ones shown in Figure 4, the last layer has a high entropy from [CLS] of 3.89 nats, indicating very broad attention. This finding makes sense given that the representation for the [CLS] token is used as input for the "next sentence prediction" task during pre-training, so it attends broadly to aggregate a representation for the whole input in the last layer.

4 Probing Individual Attention Heads

Next, we investigate individual attention heads to probe what aspects of language they have learned. In particular, we evaluate attention heads on labeled datasets for tasks like dependency parsing. An overview of our results is shown in Figure 5.

4.1 Method

We wish to evaluate attention heads at word-level tasks, but BERT uses byte-pair tokenization (Sennrich et al., 2016), which means some words (∼8% in our data) are split up into multiple tokens. We therefore convert token-token attention maps to word-word attention maps. For attention *to* a split-up word, we sum up the attention weights over its tokens. For attention *from* a split-up word, we take the mean of the attention weights over its tokens. These transformations preserve the property that the attention from each word sums to 1. For a given attention head and word, we take whichever other word receives the most attention weight as that model's prediction[3]

4.2 Dependency Syntax

Setup. We extract attention maps from BERT on the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1993) annotated with Stanford Dependencies. We evaluate both "directions" of prediction for each attention head: the head word attending to the dependent and the dependent attending to the head word. Some dependency relations are simpler to predict than others: for example a noun's determiner is often the immediately preceding word. Therefore as a point of comparison, we show predictions from a simple fixed-offset baseline. For example, a fixed offset of -2 means the word two positions to the left of the dependent is always considered to be the head.

Results. Table 1 shows that there is no single attention head that does well at syntax "overall"; the best head gets 34.5 UAS, which is not much better than the right-branching baseline, which gets 26.3 UAS. This finding is similar to the one reported by Raganato and Tiedemann (2018), who also evaluate individual attention heads for syntax.

However, we do find that certain attention heads specialize to specific dependency relations, some-

[3]We ignore [SEP] and [CLS], although in practice this does not significantly change the accuracies for most heads.

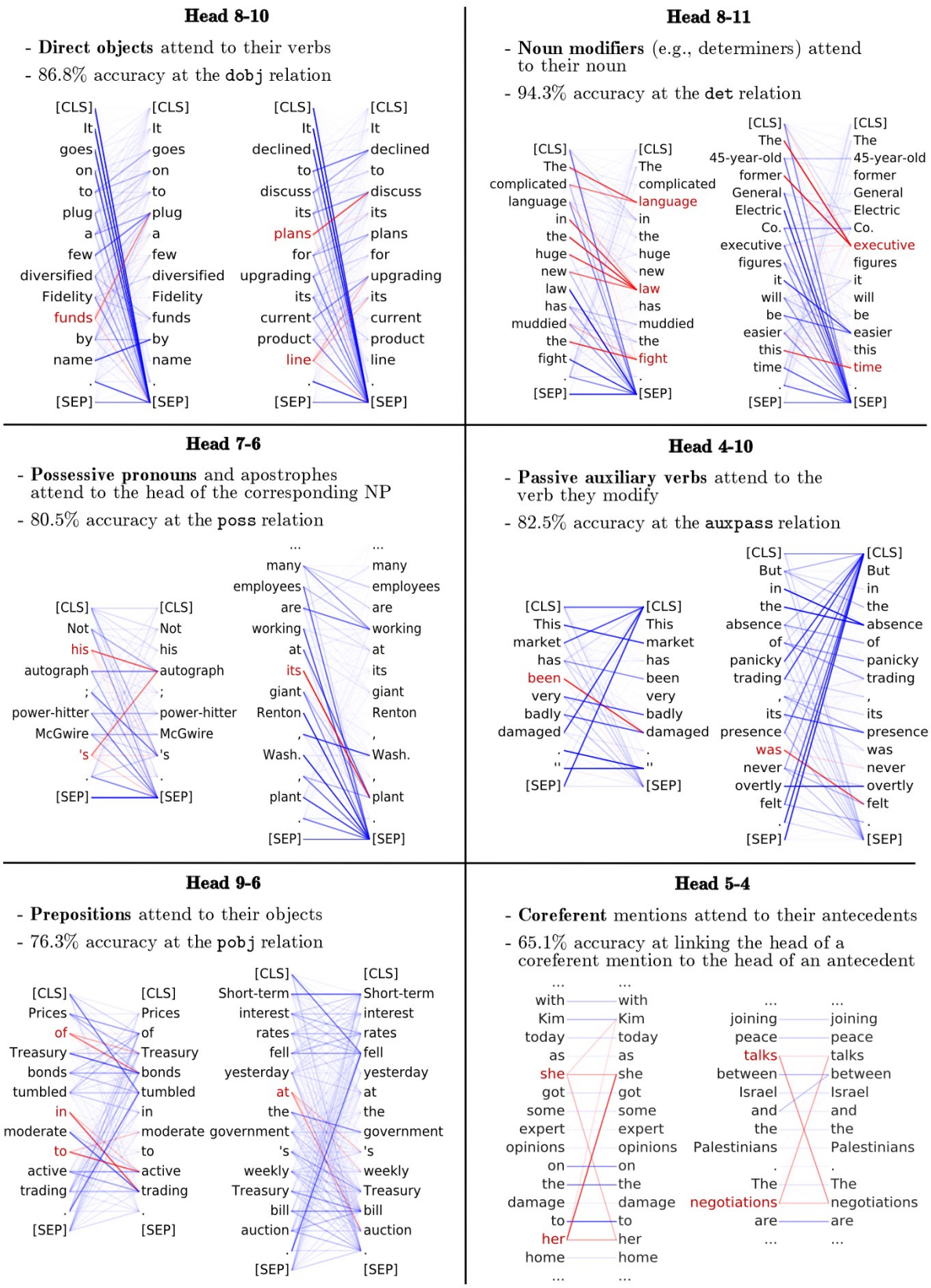

Figure 5: BERT attention heads that correspond to linguistic phenomena. In the example attention maps, the darkness of a line indicates the strength of the attention weight. All attention to/from red words is colored red; these colors are there to highlight certain parts of the attention heads' behaviors. For Head 9-6, we don't show attention to [SEP] for clarity. Despite not being explicitly trained on these tasks, BERT's attention heads perform remarkably well, illustrating how syntax-sensitive behavior can emerge from self-supervised training alone.

Relation	Head	Accuracy	Baseline
All	7-6	34.5	26.3 (1)
prep	7-4	66.7	61.8 (-1)
pobj	9-6	**76.3**	34.6 (-2)
det	8-11	**94.3**	51.7 (1)
nn	4-10	70.4	70.2 (1)
nsubj	8-2	58.5	45.5 (1)
amod	4-10	75.6	68.3 (1)
dobj	8-10	**86.8**	40.0 (-2)
advmod	7-6	48.8	40.2 (1)
aux	4-10	81.1	71.5 (1)
poss	7-6	**80.5**	47.7 (1)
auxpass	4-10	**82.5**	40.5 (1)
ccomp	8-1	**48.8**	12.4 (-2)
mark	8-2	**50.7**	14.5 (2)
prt	6-7	**99.1**	91.4 (-1)

Table 1: The best performing attentions heads of BERT on WSJ dependency parsing by dependency type. Numbers after baseline accuracies show the best offset found (e.g., (1) means the word to the right is predicted as the head). We show the 10 most common relations as well as 5 other ones attention heads do well on. Bold highlights particularly effective heads.

times achieving high accuracy and substantially outperforming the fixed-offset baseline. We find that for all relations in Table 1 except pobj, the dependent attends to the head word rather than the other way around, likely because each dependent has exactly one head but heads have multiple dependents. We also note heads can disagree with standard annotation conventions while still performing syntactic behavior. For example, head 7-6 marks *'s* as the dependent for the poss relation, while gold-standard labels mark the complement of an *'s* as the dependent (the accuracy in Table 1 counts *'s* as correct). Such disagreements highlight how these syntactic behaviors in BERT are learned as a by-product of self-supervised training, not by copying a human design.

Figure 5 shows some examples of the attention behavior. While the similarity between machine-learned attention weights and human-defined syntactic relations are striking, we note these are relations for which attention heads do particularly well on. There are many relations for which BERT only slightly improves over the simple baseline, so we would not say individual attention heads capture dependency structure as a whole. We think it would be interesting future work to extend our analysis to see if the relations well-captured by attention are similar or different for other languages.

4.3 Coreference Resolution

Having shown BERT attention heads reflect certain aspects of syntax, we now explore using attention heads for the more challenging semantic task of coreference resolution. Coreference links are usually longer than syntactic dependencies and state-of-the-art systems generally perform much worse at coreference compared to parsing.

Setup. We evaluate the attention heads on coreference resolution using the CoNLL-2012 dataset[4] (Pradhan et al., 2012). In particular, we compute antecedent selection accuracy: what percent of the time does the head word of a coreferent mention most attend to the head of one of that mention's antecedents. We compare against three baselines for selecting an antecedent:

- Picking the nearest other mention.
- Picking the nearest other mention with the same head word as the current mention.
- A simple rule-based system inspired by Lee et al. (2011). It proceeds through 4 sieves: (1) full string match, (2) head word match, (3) number/gender/person match, (4) all other mentions. The nearest mention satisfying the earliest sieve is returned.

We also show the performance of a recent neural coreference system from Wiseman et al. (2015).

Results. Results are shown in Table 2. We find that one of BERT's attention heads achieves decent coreference resolution performance, improving by over 10 accuracy points on the string-matching baseline and performing close to the rule-based system. It is particularly good with nominal mentions, perhaps because it is capable of fuzzy matching between synonyms as seen in the bottom right of Figure 5.

5 Probing Attention Head Combinations

Since individual attention heads specialize to particular aspects of syntax, the model's overall "knowledge" about syntax is distributed across multiple attention heads. We now measure this overall ability by proposing a novel family of attention-based probing classifiers and applying

[4]We truncate documents to 128 tokens long to keep memory usage manageable.

Model	All	Pronoun	Proper	Nominal
Nearest	27	29	29	19
Head match	52	47	67	40
Rule-based	69	70	77	60
Neural coref	83*	–	–	–
Head 5-4	65	64	73	58

*Only roughly comparable because on non-truncated documents and with different mention detection.

Table 2: Accuracies (%) for systems at selecting a correct antecedent given a coreferent mention in the CoNLL-2012 data. One of BERT's attention heads performs fairly well at coreference.

them to dependency parsing. For these classifiers we treat the BERT attention outputs as fixed, i.e., we do not back-propagate into BERT and only train a small number of parameters.

The probing classifiers are basically graph-based dependency parsers. Given an input word, the classifier produces a probability distribution over other words in the sentence indicating how likely each other word is to be the syntactic head of the current one.

Attention-Only Probe. Our first probe learns a simple linear combination of attention weights.

$$p(i|j) \propto \exp\left(\sum_{k=1}^{n} w_k \alpha_{ij}^k + u_k \alpha_{ji}^k\right)$$

where $p(i|j)$ is the probability of word i being word j's syntactic head, α_{ij}^k is the attention weight from word i to word j produced by head k, and n is the number of attention heads. We include both directions of attention: candidate head to dependent as well as dependent to candidate head. The weight vectors w and u are trained using standard supervised learning on the train set.

Attention-and-Words Probe. Given our finding that heads specialize to particular syntactic relations, we believe probing classifiers should benefit from having information about the input words. In particular, we build a model that sets the weights of the attention heads based on the GloVe (Pennington et al., 2014) embeddings for the input words. Intuitively, if the dependent and candidate head are "the" and "cat," the probing classifier should learn to assign most of the weight to the head 8-11, which achieves excellent performance at the determiner relation. The attention-and-words probing classifier assigns the probability of word i being word j's head as

$$p(i|j) \propto \exp\left(\sum_{k=1}^{n} W_{k,:}(v_i \oplus v_j)\alpha_{ij}^k + U_{k,:}(v_i \oplus v_j)\alpha_{ji}^k\right)$$

Where v denotes GloVe embeddings and $\oplus$ denotes concatenation. The GloVe embeddings are held fixed in training, so only the two weight matrices W and U are learned. The dot product $W_{k,:}(v_i \oplus v_j)$ produces a word-sensitive weight for the particular attention head.

Results. We evaluate our methods on the Penn Treebank dev set annotated with Stanford dependencies. We compare against three baselines:

- A right-branching baseline that always predicts the head is to the dependent's right.
- A simple one-hidden-layer network that takes as input the GloVe embeddings for the dependent and candidate head as well as distance features between the two words.[5]
- Our attention-and-words probe, but with attention maps from a BERT network with pretrained word/positional embeddings but randomly initialized other weights. This kind of baseline is surprisingly strong at other probing tasks (Conneau et al., 2018).

Results are shown in Table 3. We find the Attn + GloVe probing classifier substantially outperforms our baselines and achieves a decent UAS of 77, suggesting BERT's attention maps have a fairly thorough representation of English syntax.

As a rough comparison, we also report results from the structural probe from Hewitt and Manning (2019), which builds a probing classifier on top of BERT's vector representations rather than attention. The scores are not directly comparable because the structural probe only uses a single layer of BERT, produces undirected rather than directed parse trees, and is trained to produce the syntactic distance between words rather than directly predicting the tree structure. Nevertheless, the similarity in score to our Attn + Glove probing classifier suggests there is not much more syntactic information in BERT's vector representations compared to its attention maps.

[5]Indicator features for short distances as well as continuous distance features, with distance ahead/behind treated separately to capture word order

Model	UAS
Structural probe	80 UUAS*
Right-branching	26
Distances + GloVe	58
Random Init Attn + GloVe	30
Attn	61
Attn + GloVe	77

Table 3: Results of attention-based probing classifiers on dependency parsing. A simple model taking BERT attention maps and GloVe embeddings as input performs quite well. *Not directly comparable to our numbers; see text.

Overall, our results from probing both individual and combinations of attention heads suggest that BERT learns some aspects syntax purely as a by-product of self-supervised training. Other work has drawn a similar conclusions from examining BERT's predictions on agreement tasks (Goldberg, 2019) or internal vector representations (Hewitt and Manning, 2019; Liu et al., 2019). Traditionally, syntax-aware models have been developed through architecture design (e.g., recursive neural networks) or from direct supervision from human-curated treebanks. Our findings are part of a growing body of work indicating that indirect supervision from rich pre-training tasks like language modeling can also produce models sensitive to language's hierarchical structure.

6 Clustering Attention Heads

Are attention heads in the same layer similar to each other or different? Can attention heads be clearly grouped by behavior? We investigate these questions by computing the distances between all pairs of attention heads. Formally, we measure the distance between two heads H_i and H_j as:

$$\sum_{\text{token} \in \text{data}} JS(\mathrm{H}_i(\text{token}), \mathrm{H}_j(\text{token}))$$

Where JS is the Jensen-Shannon Divergence between attention distributions. Using these distances, we visualize the attention heads by applying multidimensional scaling (Kruskal, 1964) to embed each head in two dimensions such that the Euclidean distance between embeddings reflects the Jensen-Shannon distance between the corresponding heads as closely as possible.

Results are shown in Figure 6. We find that there are several clear clusters of heads that behave similarly, often corresponding to behaviors we have already discussed in this paper. Heads within the same layer are often fairly close to each other, meaning that heads within the layer have similar attention distributions. This finding is a bit surprising given that Tu et al. (2018) show that encouraging attention heads to have different behaviors can improve Transformer performance at machine translation. One possibility for the apparent redundancy in BERT's attention heads is the use

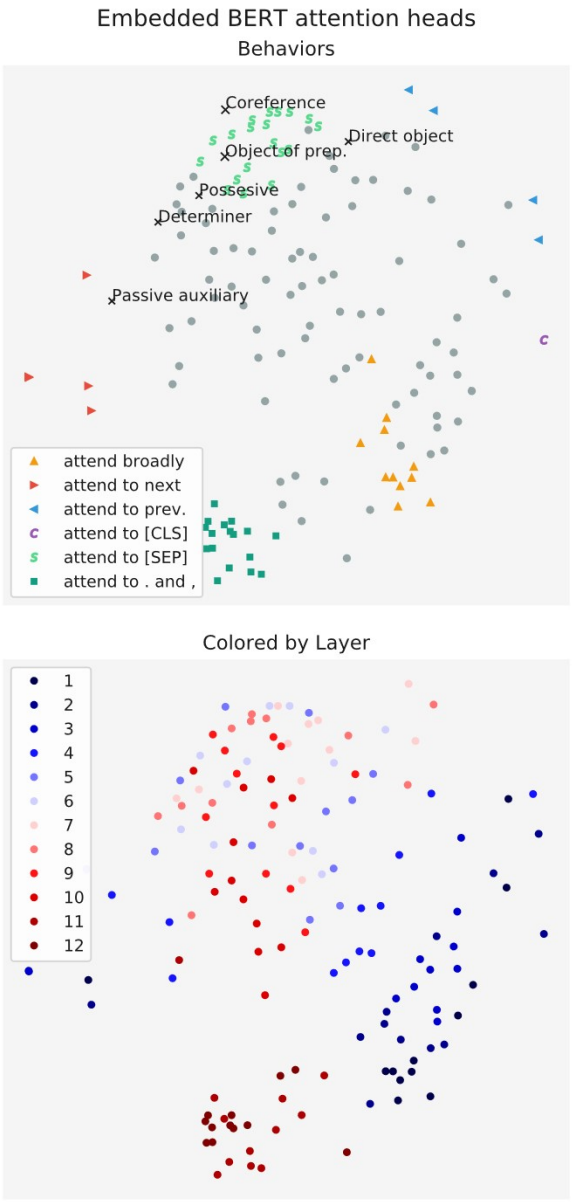

Figure 6: BERT attention heads embedded in two-dimensional space. Distance between points approximately matches the average Jensen-Shannon divergences between the outputs of the corresponding heads. Heads in the same layer tend to be close together. Attention head "behavior" was found through the analysis methods discussed throughout this paper.

of attention dropout, which causes some attention weights to be zeroed-out during training.

7 Related Work

There has been substantial recent work performing analysis to better understand what neural networks learn, especially from language model pre-training. One line of research examines the *outputs* of language models on carefully chosen input sentences (Linzen et al., 2016; Khandelwal et al., 2018; Gulordava et al., 2018; Marvin and Linzen, 2018). For example, the model's performance at subject-verb agreement (generating the correct number of a verb far away from its subject) provides a measure of the model's syntactic ability, although it does not reveal *how* that ability is captured by the network.

Another line of work investigates the internal *vector representations* of the model (Adi et al., 2017; Giulianelli et al., 2018; Zhang and Bowman, 2018), often using probing classifiers. Probing classifiers are simple neural networks that take the vector representations of a pre-trained model as input and are trained to do a supervised task (e.g., part-of-speech tagging). If a probing classifier achieves high accuracy, it suggests that the input representations reflect the corresponding aspect of language (e.g., low-level syntax). Like our work, some of these studies have also demonstrated models capturing aspects of syntax (Shi et al., 2016; Blevins et al., 2018) or coreference (Tenney et al., 2018, 2019; Liu et al., 2019) without explicitly being trained for the tasks.

With regards to analyzing attention, Vig (2019) builds a visualization tool for the BERT's attention and reports observations about the attention behavior, but does not perform quantitative analysis. Burns et al. (2018) analyze the attention of memory networks to understand model performance on a question answering dataset. There has also been some initial work in correlating attention with syntax. Raganato and Tiedemann (2018) evaluate the attention heads of a machine translation model on dependency parsing, but only report overall UAS scores instead of investigating heads for specific syntactic relations or using probing classifiers. Marecek and Rosa (2018) propose heuristic ways of converting attention scores to syntactic trees, but do not quantitatively evaluate their approach. For coreference, Voita et al. (2018) show that the attention of a context-aware neural machine translation system captures anaphora, similar to our finding for BERT.

Concurrently with our work Voita et al. (2019) identify syntactic, positional, and rare-word-sensitive attention heads in machine translation models. They also demonstrate that many attention heads can be pruned away without substantially hurting model performance. Interestingly, the important attention heads that remain after pruning tend to be ones with identified behaviors. Michel et al. (2019) similarly show that many of BERT's attention heads can be pruned. Although our analysis in this paper only found interpretable behaviors in a subset of BERT's attention heads, these recent works suggest that there might not be much to explain for some attention heads because they have little effect on model perfomance.

Jain and Wallace (2019) argue that attention often does not "explain" model predictions. They show that attention weights frequently do not correlate with other measures of feature importance. Furthermore, attention weights can often be substantially changed without altering model predictions. However, our motivation for looking at attention is different: rather than explaining model predictions, we are seeking to understand information learned by the models. For example, if a particular attention head learns a syntactic relation, we consider that an important finding from an analysis perspective even if that head is not always used when making predictions for some downstream task.

8 Conclusion

We have proposed a series of analysis methods for understanding the attention mechanisms of models and applied them to BERT. While most recent work on model analysis for NLP has focused on probing vector representations or model outputs, we have shown that a substantial amount of linguistic knowledge can be found not only in the hidden states, but also in the attention maps. We think probing attention maps complements these other model analysis techniques, and should be part of the toolkit used by researchers to understand what neural networks learn about language.

Acknowledgements

We thank the anonymous reviews for their thoughtful comments and suggestions. Kevin is supported by a Google PhD Fellowship.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *ICLR*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James R. Glass. 2017. What do neural machine translation models learn about morphology? In *ACL*.

Terra Blevins, Omer Levy, and Luke S. Zettlemoyer. 2018. Deep rnns encode soft hierarchical syntax. In *ACL*.

Kaylee Burns, Aida Nematzadeh, Alison Gopnik, and Thomas L. Griffiths. 2018. Exploiting attention to reveal shortcomings in memory models. In *BlackboxNLP@EMNLP*.

Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2018. Syntax-directed attention for neural machine translation. In *AAAI*.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *ACL*.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *NIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. In *ACL*.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem H. Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *BlackboxNLP@EMNLP*.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *NAACL-HLT*.

John Hewitt and Christopher D. Manning. 2019. Finding syntax with structural probes. In *NAACL-HLT*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Urvashi Khandelwal, He He, Peng Qi, and Daniel Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *ACL*.

Joseph B Kruskal. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *CoNLL*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *TACL*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, M. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *NAACL-HLT*.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The Penn treebank. *Computational linguistics*, 19(2):313–330.

David Marecek and Rudolf Rosa. 2018. Extracting syntactic trees from transformer encoder self-attentions. In *BlackboxNLP@EMNLP*.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *EMNLP*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *arXiv preprint arXiv:1905.10650*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

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

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *https://blog.openai.com/language-unsupervised*.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *BlackboxNLP@EMNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural mt learn source syntax? In *EMNLP*.

Emma Strubell, Patrick Verga, Daniel Andor, David I Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *EMNLP*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *ICML*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2018. What do you learn from context? probing for sentence structure in contextualized word representations. In *ICLR*.

Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-head attention with disagreement regularization. In *EMNLP*.

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 *NIPS*.

Jesse Vig. 2019. Visualizing attention in transformer-based language models. *arXiv preprint arXiv:1904.02679*.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *ACL*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Sam Joshua Wiseman, Alexander Matthew Rush, Stuart Merrill Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *ACL*.

Kelly W. Zhang and Samuel R. Bowman. 2018. Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis. In *BlackboxNLP@EMNLP*.