BERT Rediscovers the Classical NLP Pipeline

Ian Tenney[1]   **Dipanjan Das**[1]   **Ellie Pavlick**[1,2]
[1]Google Research   [2]Brown University
{iftenney,dipanjand,epavlick}@google.com

Abstract

Pre-trained text encoders have rapidly advanced the state of the art on many NLP tasks. We focus on one such model, BERT, and aim to quantify where linguistic information is captured within the network. We find that the model represents the steps of the traditional NLP pipeline in an interpretable and localizable way, and that the regions responsible for each step appear in the expected sequence: POS tagging, parsing, NER, semantic roles, then coreference. Qualitative analysis reveals that the model can and often does adjust this pipeline dynamically, revising lower-level decisions on the basis of disambiguating information from higher-level representations.

1 Introduction

Pre-trained sentence encoders such as ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2019) have rapidly improved the state of the art on many NLP tasks, and seem poised to displace both static word embeddings (Mikolov et al., 2013) and discrete pipelines (Manning et al., 2014) as the basis for natural language processing systems. While this has been a boon for performance, it has come at the cost of interpretability, and it remains unclear whether such models are in fact learning the kind of abstractions that we intuitively believe are important for representing natural language, or are simply modeling complex co-occurrence statistics.

A wave of recent work has begun to "probe" state-of-the-art models to understand whether they are representing language in a satisfying way. Much of this work is behavior-based, designing controlled test sets and analyzing errors in order to reverse-engineer the types of abstractions the model may or may not be representing (e.g. Conneau et al., 2018; Marvin and Linzen, 2018; Poliak et al., 2018). Parallel efforts inspect the structure

of the network directly, to assess whether there exist localizable regions associated with distinct types of linguistic decisions. Such work has produced evidence that deep language models can encode a range of syntactic and semantic information (e.g. Shi et al., 2016; Belinkov, 2018; Tenney et al., 2019), and that more complex structures are represented hierarchically in the higher layers of the model (Peters et al., 2018b; Blevins et al., 2018).

We build on this latter line of work, focusing on the BERT model (Devlin et al., 2019), and use a suite of probing tasks (Tenney et al., 2019) derived from the traditional NLP pipeline to quantify where specific types of linguistic information are encoded. Building on observations (Peters et al., 2018b) that lower layers of a language model encode more local syntax while higher layers capture more complex semantics, we present two novel contributions. First, we present an analysis that spans the common components of a traditional NLP pipeline. We show that the order in which specific abstractions are encoded reflects the traditional hierarchy of these tasks. Second, we qualitatively analyze how individual sentences are processed by the BERT network, layer-by-layer. We show that while the pipeline order holds in aggregate, the model can allow individual decisions to depend on each other in arbitrary ways, deferring ambiguous decisions or revising incorrect ones based on higher-level information.

2 Model

Edge Probing. Our experiments are based on the "edge probing" approach of Tenney et al. (2019), which aims to measure how well information about linguistic structure can be extracted from a pre-trained encoder. Edge probing decomposes structured-prediction tasks into a common

format, where a probing classifier receives spans $\mathbf{s}_1 = [i_1, j_1)$ and (optionally) $\mathbf{s}_2 = [i_2, j_2)$ and must predict a label such as a constituent or relation type.[1] The probing classifier has access only to the per-token contextual vectors *within* the target spans, and so must rely on the encoder to provide information about the relation between these spans and their role in the sentence.

We use eight labeling tasks from the edge probing suite: part-of-speech (POS), constituents (Consts.), dependencies (Deps.), entities, semantic role labeling (SRL), coreference (Coref.), semantic proto-roles (SPR; Reisinger et al., 2015), and relation classification (SemEval). These tasks are derived from standard benchmark datasets, and evaluated with a common metric–micro-averaged F1–to facilitate comparison across tasks. [2]

BERT. The BERT model (Devlin et al., 2019) has shown state-of-the-art performance on many tasks, and its deep Transformer architecture (Vaswani et al., 2017) is typical of many recent models (e.g. Radford et al., 2018, 2019; Liu et al., 2019). We focus on the stock BERT models (base and large, uncased), which are trained with a multi-task objective (masked language modeling and next-sentence prediction) over a 3.3B word English corpus. Since we want to understand how the network represents language as a result of pre-training, we follow Tenney et al. (2019) (departing from standard BERT usage) and freeze the encoder weights. This prevents the encoder from rearranging its internal representations to better suit the probing task.

Given input tokens $T = [t_0, t_1, \ldots, t_n]$, a deep encoder produces a set of layer activations $H^{(0)}, H^{(1)}, \ldots, H^{(L)}$, where $H^{(\ell)} = [\mathbf{h}_0^{(\ell)}, \mathbf{h}_1^{(\ell)}, \ldots, \mathbf{h}_n^{(\ell)}]$ are the activation vectors of the ℓth encoder layer and $H^{(0)}$ corresponds to the non-contextual word(piece) embeddings. We use a weighted sum across layers (§3.1) to pool these into a single set of per-token representation vectors $H = [\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_n]$, and train a probing classifier P_τ for each task using the architecture and procedure of Tenney et al. (2019).

---

[1] For single-span tasks (POS, entities, and constituents), $\mathbf{s}_2$ is not used. For POS, $\mathbf{s}_1 = [i, i + 1)$ is a single token.

[2] We use the code from https://github.com/jsalt18-sentence-repl/jiant. Dependencies is the English Web Treebank (Silveira et al., 2014), SPR is the SPR1 dataset of (Teichert et al., 2017), and relations is SemEval 2010 Task 8 (Hendrickx et al., 2009). All other tasks are from OntoNotes 5.0 (Weischedel et al., 2013).

Limitations This work is intended to be exploratory. We focus on one particular encoder–BERT–to explore how information can be organized in a deep language model, and further work is required to determine to what extent the trends hold in general. Furthermore, our work carries the limitations of all inspection-based probing: the fact that a linguistic pattern is not observed by our probing classifier does not guarantee that it is not there, and the observation of a pattern does not tell us how it is used. For this reason, we emphasize the importance of combining structural analysis with behavioral studies (as discussed in § 1) to provide a more complete picture of what information these models encode and how that information affects performance on downstream tasks.

3 Metrics

We define two complementary metrics. The first, scalar mixing weights (§3.1) tell us which layers, in combination, are most relevant when a probing classifier has access to the whole BERT model. The second, cumulative scoring (§3.2) tells us how much higher we can score on a probing task with the introduction of each layer. These metrics provide complementary views on what is happening inside the model. Mixing weights are learned solely from the training data–they tell us which layers the probing model finds most useful. In contrast, cumulative scoring is derived entirely from an evaluation set, and tell us how many layers are needed for a correct prediction.

3.1 Scalar Mixing Weights

To pool across layers, we use the scalar mixing technique introduced by the ELMo model. Following Equation (1) of Peters et al. (2018a), for each task we introduce scalar parameters γ_τ and $a_\tau^{(0)}, a_\tau^{(1)}, \ldots, a_\tau^{(L)}$, and let:

$$\mathbf{h}_{i,\tau} = \gamma_\tau \sum_{\ell=0}^{L} s_\tau^{(\ell)} \mathbf{h}_i^{(\ell)} \qquad (1)$$

where $\mathbf{s}_\tau = \text{softmax}(\mathbf{a}_\tau)$. We learn these weights jointly with the probing classifier P_τ, in order to allow it to extract information from the many layers of an encoder without adding a large number of parameters. After the probing model is trained, we extract the learned coefficients in order to estimate the contribution of different layers to that particular task. We interpret higher weights as ev-

4594

Figure 1: Summary statistics on BERT-large. Columns on left show F1 dev-set scores for the baseline ($P_\tau^{(0)}$) and full-model ($P_\tau^{(L)}$) probes. Dark (blue) are the mixing weight center of gravity (Eq. 2); light (purple) are the expected layer from the cumulative scores (Eq. 4).

idence that the corresponding layer contains more information related to that particular task.

Center-of-Gravity. As a summary statistic, we define the mixing weight center of gravity as:

$$\bar{E}_s[\ell] = \sum_{\ell=0}^{L} \ell \cdot s_\tau^{(\ell)} \qquad (2)$$

This reflects the average layer attended to for each task; intuitively, we can interpret a higher value to mean that the information needed for that task is captured by higher layers.

3.2 Cumulative Scoring

We would like to estimate at which layer in the encoder a target ($\mathtt{s_1}, \mathtt{s_2}, \mathtt{label}$) can be correctly predicted. Mixing weights cannot tell us this directly, because they are learned as parameters and do not correspond to a distribution over data. A naive classifier at a single layer cannot either, because information about a particular span may be spread out across several layers, and as observed in Peters et al. (2018b) the encoder may choose to discard information at higher layers.

To address this, we train a series of classifiers $\{P_\tau^{(\ell)}\}_\ell$ which use scalar mixing (Eq. 1) to attend to layer ℓ as well as *all previous* layers. $P_\tau^{(0)}$ corresponds to a non-contextual baseline that uses only a bag of word(piece) embeddings, while $P_\tau^{(L)} = P_\tau$ corresponds to probing all layers of the BERT model.

These classifiers are cumulative, in the sense that $P_\tau^{(\ell+1)}$ has a similar number of parameters but with access to strictly more information than $P_\tau^{(\ell)}$,
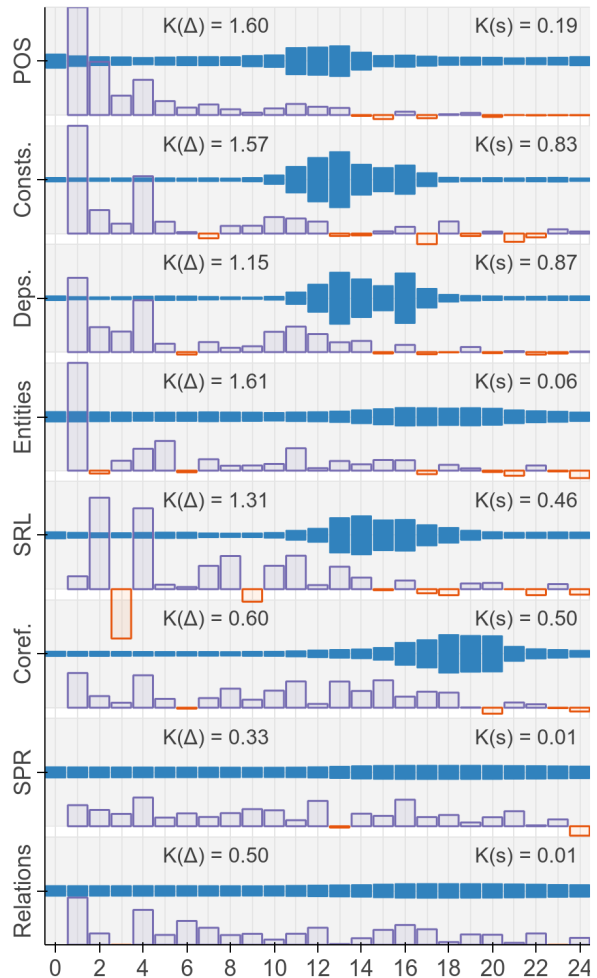

Figure 2: Layer-wise metrics on BERT-large. Solid (blue) are mixing weights $s_\tau^{(\ell)}$ (§3.1); outlined (purple) are differential scores $\Delta_\tau^{(\ell)}$ (§3.2), normalized for each task. Horizontal axis is encoder layer.

and we see intuitively that performance (F1 score) generally increases as more layers are added.[3] We can then compute a differential score $\Delta_\tau^{(\ell)}$, which measures how much better we do on the probing task if we observe one additional encoder layer ℓ:

$$\Delta_\tau^{(\ell)} = \mathrm{Score}(P_\tau^{(\ell)}) - \mathrm{Score}(P_\tau^{(\ell-1)}) \qquad (3)$$

Expected Layer. Again, we compute a (pseudo)[4] expectation over the differential scores as a summary statistic. To focus on the behavior of the contextual encoder layers, we omit the contribution of both the "trivial" examples resolved at layer 0, as well as the remaining headroom from

---

[3]Note that if a new layer provides distracting features, the probing model can overfit and performance can drop. We see this in particular in the last 1-2 layers (Figure 2).

[4]This is not a true expectation because the F1 score is not an expectation over examples.

the full model. Let:

$$\bar{E}_\Delta[\ell] = \frac{\sum_{\ell=1}^{L} \ell \cdot \Delta_\tau^{(\ell)}}{\sum_{\ell=1}^{L} \Delta_\tau^{(\ell)}} \quad (4)$$

This can be thought of as, approximately, the expected layer at which the probing model correctly labels an example, assuming that example is resolved at *some* layer $\ell \geq 1$ of the encoder.

4   Results

Figure 1 reports summary statistics and absolute F1 scores, and Figure 2 reports per-layer metrics. Both show results on the 24-layer BERT-large model. We also report $K(\star) = \mathrm{KL}(\star||\mathrm{Uniform})$ to estimate how non-uniform[5] each statistic ($\star = s_\tau, \Delta_\tau$) is for each task.

Linguistic Patterns.   We observe a consistent trend across both of our metrics, with the tasks encoded in a natural progression: POS tags processed earliest, followed by constituents, dependencies, semantic roles, and coreference. That is, it appears that basic syntactic information appears earlier in the network, while high-level semantic information appears at higher layers. We note that this finding is consistent with initial observations by Peters et al. (2018b), which found that constituents are represented earlier than coreference.

In addition, we observe that in general, syntactic information is more localizable, with weights related to syntactic tasks tending to be concentrated on a few layers (high $K(s)$ and $K(\Delta)$), while information related to semantic tasks is generally spread across the entire network. For example, we find that for semantic relations and proto-roles (SPR), the mixing weights are close to uniform, and that nontrivial examples for these tasks are resolved gradually across nearly all layers. For entity labeling many examples are resolved in layer 1, but with a long tail thereafter, and only a weak concentration of mixing weights in high layers. Further study is needed to determine whether this is because BERT has difficulty representing the correct abstraction for these tasks, or because semantic information is inherently harder to localize.

Comparison of Metrics.   For many tasks, we find that the differential scores are highest in the first few layers of the model (layers 1-7 for BERT-large), i.e. most examples can be correctly classified very early on. We attribute this to the availability of heuristic shortcuts: while challenging examples may not be resolved until much later, many cases can be guessed from shallow statistics. Conversely, we observe that the learned mixing weights are concentrated much later, layers 9-20 for BERT-large. We observe–particularly when weights are highly concentrated–that the highest weights are found on or just after the *highest* layers which give an improvement $\Delta_\tau^{(\ell)}$ in F1 score for that task.

This helps explain the observations on the semantic relations and SPR tasks: cumulative scoring shows continued improvement up to the highest layers of the model, while the lack of concentration in the mixing weights suggest that the BERT encoder does not expose a localized set of features that encode these more semantic phenomena. Similarly for entity types, we see continued improvements in the higher layers – perhaps related to fine-grained semantic distinctions like "Organization" (ORG) vs. "Geopolitical Entity" (GPE) – while the low value for the *expected* layer reflects that many examples require only limited context to resolve.

Comparison of Encoders.   We observe the same general ordering on the 12-layer BERT-base model (Figure A.2). In particular, there appears to be a "stretching effect", where the representations for a given task tend to concentrate at the same layers *relative to the top of the model*; this is illustrated side-by-side in Figure A.3.

4.1   Per-Example Analysis

We explore, qualitatively, how beliefs about the structure of individual sentences develop over the layers of the BERT network. The OntoNotes development set contains annotations for five of our probing tasks: POS, constituents, entities, SRL, and coreference. We compile the predictions of the per-layer classifiers $P_\tau^{(\ell)}$ for each task. Because many annotations are uninteresting – for example, 89% of part-of-speech tags are correct at layer 0 – we use a heuristic to identify ambiguous sentences to visualize.[6] Figure 3 shows two

---

[5]$\mathrm{KL}(\star||\mathrm{Uniform}) = -H(\star) + \mathrm{Constant}$, so higher values correspond to lower entropy.

[6]Specifically, we look for target edges (s_1, s_2, label) where the highest scoring label has an average score $\frac{1}{L+1} \sum_{\ell=0}^{L} P_\tau^{(\ell)}(\mathtt{label}|s_1, s_2) \leq 0.7$, and look at sentences with more than one such edge.

(a) he smoked **toronto** in the playoffs with six hits, seven walks and eight stolen bases ...
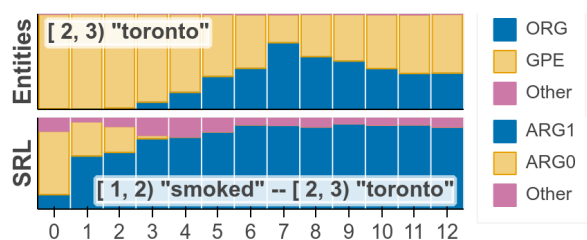
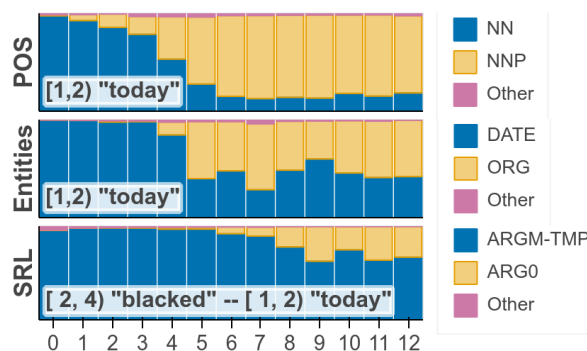(b) china **today** blacked out a cnn interview that was ...

Figure 3: Probing classifier predictions across layers of BERT-base. Blue is the correct label; orange is the incorrect label with highest average score over layers. Bar heights are (normalized) probabilities $P_\tau^{(\ell)}(\texttt{label}|\texttt{s}_1, s_2)$. In the interest of space, only selected annotations are shown.

selected examples, and more are presented in Appendix A.2.

We find that while the pipeline order holds on average (Figure 2), for individual examples the model is free to and often does choose a different order. In the first example, the model originally (incorrectly) assumes that *"Toronto"* refers to the city, tagging it as a GPE. However, after resolving the semantic role – determining that *"Toronto"* is the thing getting *"smoked"* (ARG1) – the entity-typing decision is revised in favor of ORG (i.e. the sports team). In the second example, the model initially tags *"today"* as a common noun, date, and temporal modifier (ARGM-TMP). However, this phrase is ambiguous, and it later reinterprets *"china today"* as a proper noun (i.e. the TV network) and updates its beliefs about the entity type (to ORG), followed by the semantic role (reinterpreting it as the agent ARG0).

5 Conclusion

We employ the edge probing task suite to explore how the different layers of the BERT network can resolve syntactic and semantic structure within a sentence. We present two complementary measurements: scalar mixing weights, learned from a training corpus, and cumulative scoring, measured on an evaluation set, and show that a consistent ordering emerges. We find that while this traditional pipeline order holds in the aggregate, on individual examples the network can resolve out-of-order, using high-level information like predicate-argument relations to help disambiguate low-level decisions like part-of-speech. This provides new evidence corroborating that deep language models can represent the types of syntactic and semantic abstractions traditionally believed necessary for language processing, and moreover that they can model complex interactions between different levels of hierarchical information.

References

Yonatan Belinkov. 2018. *On internal language representations in deep learning: An analysis of machine translation and speech recognition*. Ph.D. thesis, Massachusetts Institute of Technology.

Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs encode soft hierarchical syntax. In *Proceedings of ACL*.

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 *Proceedings of ACL*.

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

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint 1901.11504*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky.

2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL: System Demonstrations*.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*.

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

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of EMNLP*.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of EMNLP*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *https://blog.openai.com/better-language-models*.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic proto-roles. *Transactions of the Association of Computational Linguistics*.

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

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*.

Adam Teichert, Adam Poliak, Benjamin Van Durme, and Matthew Gormley. 2017. Semantic proto-role labeling. In *Proceedings of AAAI*.

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 *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. OntoNotes release 5.0 LDC2013T19. *Linguistic Data Consortium, Philadelphia, PA*.

A   Appendix

A.1   Comparison of Encoders

We reproduce Figure 1 and Figure 2 (which depict metrics on BERT-large) from the main paper below, and show analogous plots for the BERT-base models. We observe that the most important layers for a given task appear in roughly the same *relative* position on both the 24-layer BERT-large and 12-layer BERT-base models, and that tasks generally appear in the same order.

Additionally, in Figure A.1 we show scalar mixing weights for the ELMo encoder (Peters et al., 2018a), which consists of two LSTM layers over a per-word character CNN. We observe that the first LSTM layer (layer 1) is most informative for all tasks, which corroborates the observations of Figure 2 of Peters et al. (2018a). As with BERT, we observe that the weights are only weakly concentrated for the relations and SPR tasks. However, unlike BERT, we see only a weak concentration in the weights on the coreference task, which agrees with the finding of Tenney et al. (2019) that ELMo presents only weak features for coreference.

A.2   Additional Examples

We provide additional examples in the style of Figure 3, which illustrate sequential decisions in the layers of the BERT-base model.
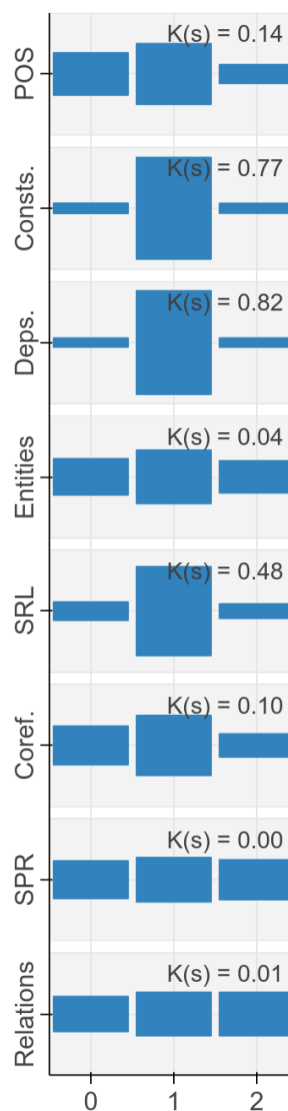

Figure A.1: Scalar mixing weights for the ELMo encoder. Layer 0 is the character CNN that produces per-word representations, and layers 1 and 2 are the LSTM layers.
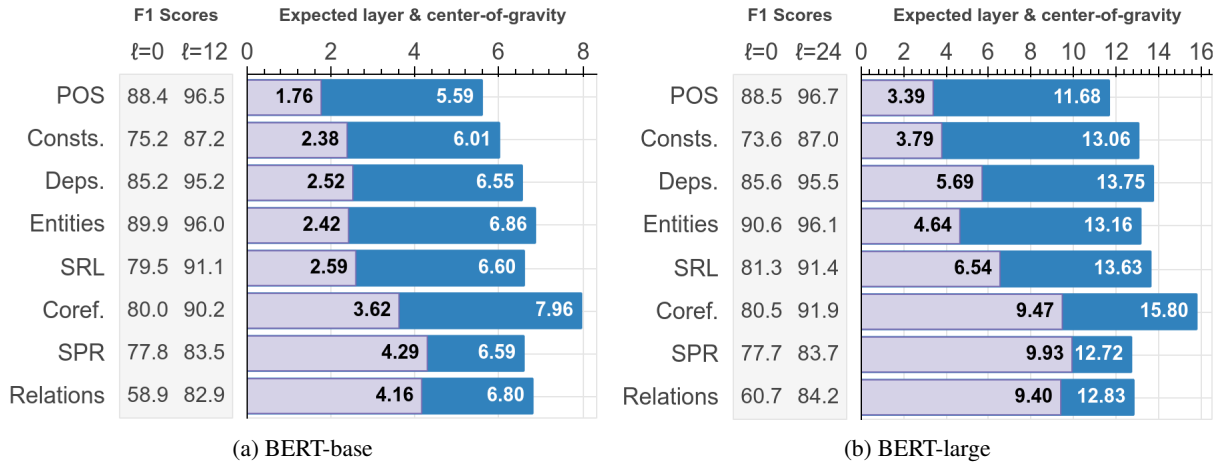
4599

Figure A.2: Summary statistics on BERT-base (left) and BERT-large (right). Columns on left show F1 dev-set scores for the baseline ($P_\tau^{(0)}$) and full-model ($P_\tau^{(L)}$) probes. Dark (blue) are the mixing weight center of gravity; light (purple) are the expected layer from the cumulative scores.
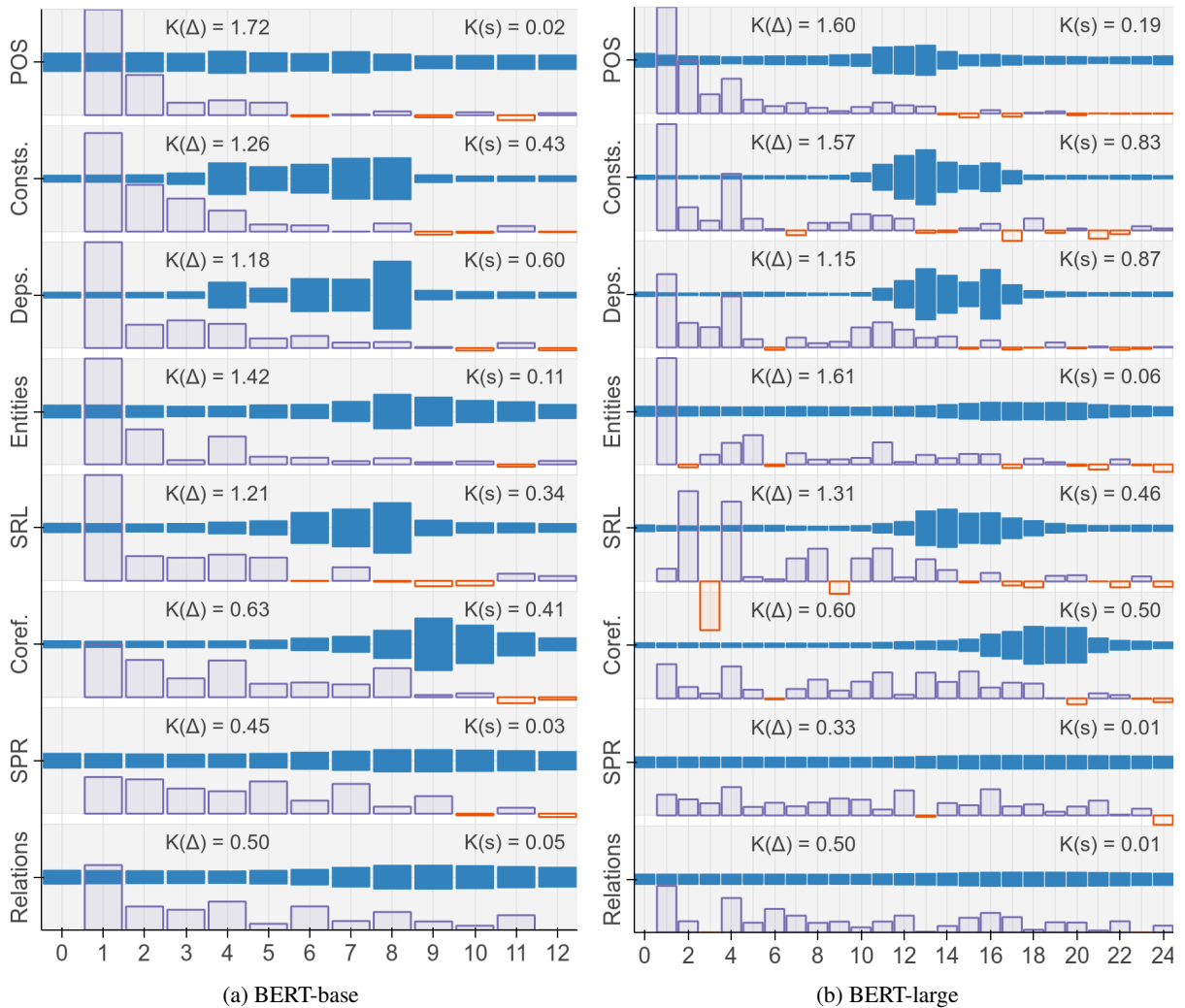

Figure A.3: Layer-wise metrics on BERT-base (left) and BERT-large (right). Solid (blue) are mixing weights $s_\tau^{(\ell)}$; outlined (purple) are differential scores $\Delta_\tau^{(\ell)}$, normalized for each task. Horizontal axis is encoder layer.

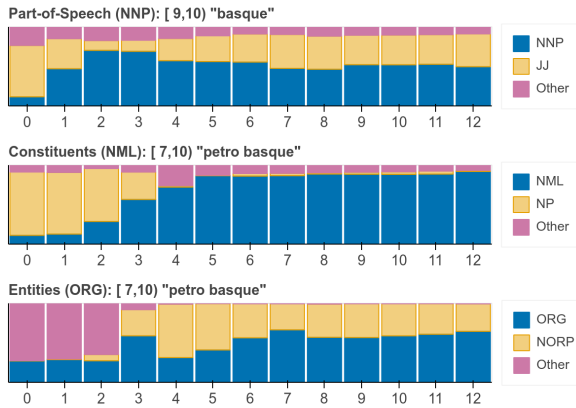saturday ' s incident was the sixth petro **basque** spill this year .

Part-of-Speech (NNP): [9,10) "basque"

Constituents (NML): [7,10) "petro basque"

Entities (ORG): [7,10) "petro basque"

Figure A.4: Trace of selected annotations that intersect the token "**basque**" in the above sentence. We see the model recognize this as part of a proper noun (NNP) in layer 2, which leads it to revise its hypothesis about the constituent "petro basque" from an ordinary noun phrase (NP) to a nominal mention (NML) in layers 3-4. We also see that from layer 3 onwards, the model recognizes "petro basque" as either an organization (ORG) or a national or religious group (NORP), but does not strongly disambiguate between the two.

he saw the hurt man , but he went around him .

OntoNotes Coref. (0): [10,11) "him" -- [7, 8) "he"

OntoNotes Coref. (0): [10,11) "him" -- [0, 1) "he"

OntoNotes Coref. (0): [2, 5) "the hurt man" -- [7, 8) "he"

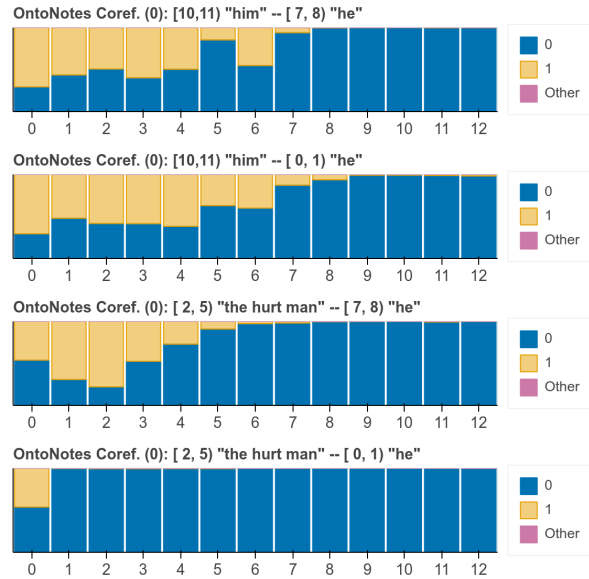**OntoNotes Coref. (0): [2, 5) "the hurt man" -- [0, 1) "he"**

Figure A.6: Trace of selected coreference annotations on the above sentence. Not shown are two coreference edges that the model has correctly resolved at layer 0 (guessing from embeddings alone): "him" and "the hurt man" are coreferent, as are "he" and "he". We see that the remaining edges, between non-coreferent mentions, are resolved in several stages.

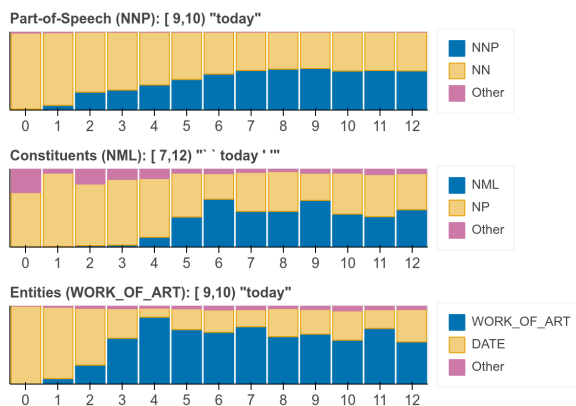today he appeared on nbc ' s `` **today** ' ' show and said he was sorry .

Part-of-Speech (NNP): [9,10) "today"

Constituents (NML): [7,12) "`` today ' '"

Entities (WORK_OF_ART): [9,10) "today"

Figure A.5: Trace of selected annotations that intersect the second "**today**" in the above sentence. The odel initially believes this to be a date and a common noun, but by layer 4 realizes that this is the TV show (entity tag WORK_OF_ART) and subsequently revises its hypotheses about the constituent type and part-of-speech.

he would not stop to help him either .

SRL (ARGM-PRP): [3, 4) "stop" -- [4, 7) "to help him"

SRL (ARG1): [5, 6) "help" -- [6, 7) "him"
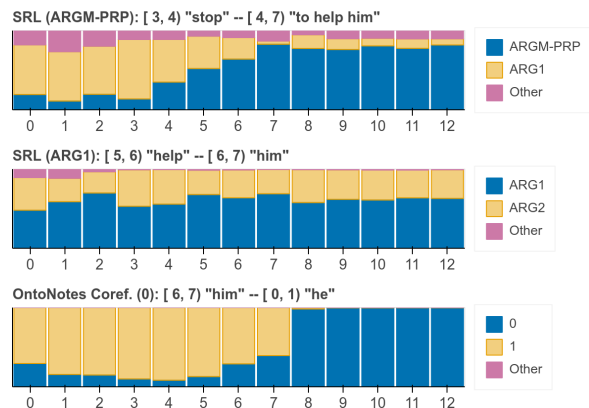
OntoNotes Coref. (0): [6, 7) "him" -- [0, 1) "he"

Figure A.7: Trace of selected coreference and SRL annotations on the above sentence. The model resolves the semantic role (purpose, ARGM-PRP) of the phrase "to help him" in layers 5-7, then quickly resolves at layer 8 that "him" and "he" (the agent of "stop") are not coreferent. Also shown is the correct prediction that "him" is the recipient (ARG1, patient) of "help".