End-to-End Model Probing

Yunxuan Xiao*†, Ashley Wu†, Su Park†, Divija Nagaraju†, Graham Neubig†
Anyscale Inc.*, Carnegie Mellon University†
yunxuanx@anyscale.com, wangchew, suminpar, dngaraj, gneubig@cs.cmu.edu

Abstract

Model probing is widely used to investigate which linguistic properties are readily accessible in the embeddings of pretrained language models. However, does that necessarily mean that the information encoded in model representation is actually useful for end tasks? In this work, we propose an empirical framework for *end-to-end model probing*, which probes the model to analyze the correlations between performance on probing and end-tasks on a system- and example-level. We probe multiple linguistic properties as probing tasks and find that (1) models with a better understanding of the sentence structure and proper nouns can better identify the entailment relations between two sentences (2) successfully detecting the semantic relationships in the higher layers is crucial in more complex tasks like paraphrase detection and natural language inference.

1 Introduction

When a child learns a language, they'll most likely start by picking up a few words from their caregivers. Will they immediately be able to say a simple sentence or two? Perhaps, yes. Will they now be able to read lengthy passages to answer questions or draw inferences about the information contained therein? This is much more unlikely. If the second task seems quite implausible, what specific knowledge would the child need beyond substantial vocabulary and rudimentary syntactic structures?

Model probing is a widely-studied field in the area of NLP (Belinkov, 2022; Ousidhoum et al., 2021; Jin et al., 2019; Hohman et al., 2019), where researchers examine whether certain linguistic competencies (e.g. basic syntax and semantic understanding) are encoded in the representations of a language model trained on vast amounts of language data. The current probing paradigm has achieved this by training a classifier for specific linguistic properties such as semantic tagging, syntactic chunking, or part of speech tagging, and evaluating its performance to determine whether a model has encoded the information necessary to easily surface these properties. However, while a classifier's performance can confirm that the model contains information about a specific linguistic property, it cannot validate whether that linguistic competency is actually used in a end task such as sentiment analysis, semantic similarity calculation, paraphrasing, or inference. For instance, if the model were to calculate the semantic similarity of the following two sentences, *"he ate salad with a fork"* and *"he ate salad with a dressing,"* we want to know if the model is able to tell whether he had also eaten the fork or the dressing along with the salad through grasping the intrinsic semantic differences between the two sentences (further detailed in Section 2).

Going back to our analogy of a child above, we want to investigate whether mastery of simple tasks (i.e., understanding vocabulary and grammar) is truly necessary for a language model, as it is for a child, to competently solve complex tasks (i.e., reading comprehension). We hypothesize that (1) nontrivial correlations between probing tasks and end tasks exist (2) different end tasks focus on different linguistic abilities (3) end-to-end probing can help deductively diagnose the causes of errors in the end tasks by attributing the success or failure of the probing tasks.

We present a model-agnostic end-to-end model probing framework[1], which generates pseudo-labels for probing tasks and uses them to probe a pre-trained encoder. At the system level, our predictions for both the end-task and various probing tasks showed that different end tasks require different probing task information, while on the instance level for each end-task, fine-tuning the pre-trained encoder for each end task enabled fine-grained analyses which shed light on the specific

[1]We release our code and data in https://github.com/neulab/end2end-probing.

linguistic properties each particular end task relies on. We aim to enable researchers to diagnose the strengths and weaknesses not only for a system but also on a single instance, interpret relationships between multiple systems, and examine prediction results to help us to better understand how models learn and how they can be improved.

2 Background

Probing Task and End Task Within the context of our work, probing tasks are those that involve more simple and fundamental operations in natural language processing such as part-of-speech and named entity recognition. End tasks are those that entail more complex operations such as sentiment analysis, text entailment, sentence similarity or equivalence, and grammaticality.

Probing Probing is a method for examining whether certain linguistic competencies (e.g. basic syntax and semantic roles) are encoded in large language models such as BERT (Devlin et al., 2018). Specifically, edge probing (Tenney et al., 2019b) measures the encoding of linguistic information in large pretrained language models by decomposing each structured task into a set of graph edges that can be predicted independently using a common classifier architecture. In an edge probing experiment, a language model is trained on a specific text corpus, then tested on a series of tasks to evaluate its performance to assess how the model has encoded the sentence structure across a range of syntactic, semantic, local, and longer-range phenomena.

Adversarial Attacks Adversarial Training (AT) has been used as a method to evaluate a model's susceptibility to purposely designed incorrect examples and improve robustness across various tasks. For text classification, Ebrahimi et al. applied white-box adversarial training on the Stanford Sentiment Treebank (SST) dataset to trick the model's character-level neural classifier with an atomic flip operation that swaps tokens based on the gradients of the one-hot input vectors. Instead of relying on an automatic example generator, Yin et al. manually collected grammatical errors made by non-native speakers to simulate adversarial attacks on clean text data to diagnose the degree of impact across different tasks including Entailment, Named Entity Recognition, and Sentiment Analysis. The authors also devised a custom linguistic acceptability task which revealed the model's abilities in identifying grammatically incorrect sentences and the position of errors. For Part-of-Speech Tagging, Yasunaga et al. applied adversarial training on the Penn Treebank WSJ corpus and the Universal Dependencies (UD) dataset across 27 languages to see improved tagging accuracy for rare words and for low-resource languages and an indication of the improved tagging performance contributing to the end task of dependency parsing.

Evaluation Methods CheckList (Ribeiro et al., 2020) and Explainaboard (Liu et al., 2021) are the two recently developed methods for interpretable evaluations the former being a task-agnostic methodology for testing NLP models across a matrix of general linguistic capabilities that facilitate comprehensive test ideation and the latter achieving the same objective with broader coverage of 400 systems, 50 datasets, 40 languages, and 12 tasks. ExplainaBoard (Liu et al., 2021) introduced the method of bucketing, which partitions results into different groups based on the defined features, visualizes the test samples' performance with respect to each bucket, and allows the users to see the corresponding errors. Despite its novel approach, the ExplainaBoard remains a static visualization tool rather than a fully-functioning, versatile interface that processes the data across different tasks, integrates mainstream probing paradigms, and evaluates language competency on a real-time basis. We aim to fill this gap with a front-end web application portal that conducts interactive data analysis, provides interactive data analysis, and accepts model submissions.

3 End-to-End Probing

3.1 Sandbox vs End-to-End Probing

One major limitation of the existing probing methods is that they probe models using out-of-domain data, which results in an inherent disconnect between the probing task and the end task. As an example, (Tenney et al., 2019a) probed the OntoNotes datasets and founded the classical NLP pipeline for QA within BERT. While the probing tasks had learned the OntoNotes data, the models had learned different task-specific data while finetuning for the end task (QA). We refer to this type of probing that dominates the current paradigm as *sandbox probing*.

While sandbox probing may allow researchers

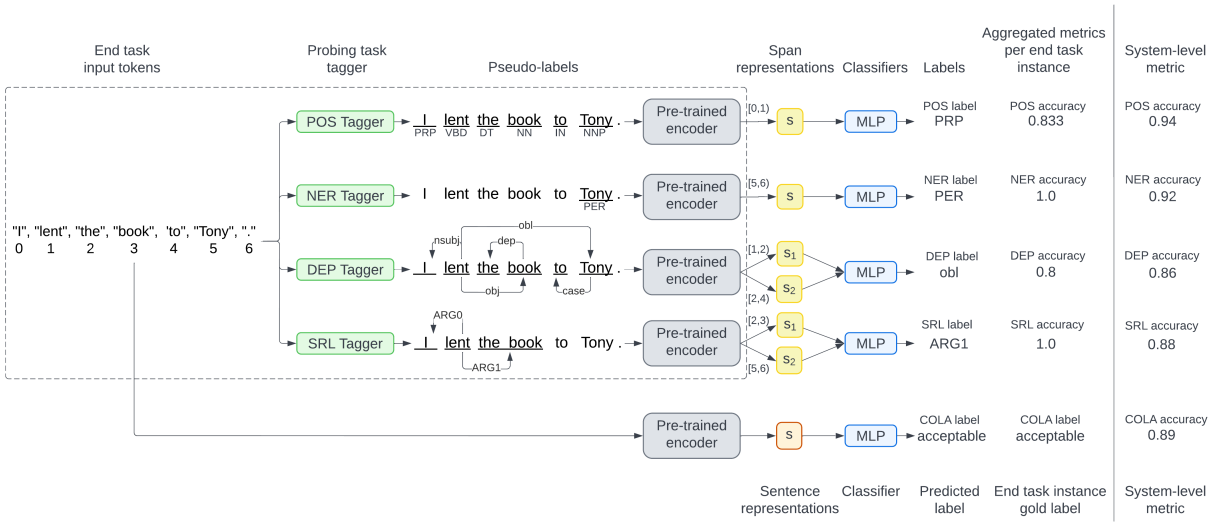

Figure 1: End-to-end model probing framework with CoLA as end task example. Parameters inside the dashed line area are fixed. Multiple pseudo-labels are generated for a given end task instance using the corresponding taggers. For PoS and NER, only a single span is used for one label. For DEP and SRL, two spans are used. We apply edge probing and train separate multi-layer perceptrons (MLP) for each probing task using span representations as input to predict labels. Since there might be more than one pseudo-labels for one end task instance, we aggregate the metric for each probing task as instance-level metrics. For the end task, we fine-tune the pretrained encoder and use sentence representation to predict the end task label. We then aggregate all the case-level metrics and predictions to obtain a system-level metric.

to detect the presence of a linguistic property in the embeddings, it does not shed light on whether that property was actually used in the end task. There has been no empirical study that focused on the utilization of the found linguistic feature to our knowledge. To bridge this gap, our work proposes *end-to-end probing*, which uses in-domain data to probe models by generating pseudo-labels of end tasks as probing datasets. The following are our research questions: *(i) Which linguistic tendencies encoded in pretrained model are helpful for a certain end task? (ii) Are there meaningful correlations between the edge probing results for probing tasks and end tasks? (iii) What are the commonalities across sentences that perform well/not well across the majority of tasks?*

3.2 End-to-End Model Probing Framework

Our end-to-end probing framework consists of the following steps: (1) We use the input tokens of an end task instance as inputs for the probing task tagger to generate pseudo-labels. (2) The pseudo-labels are then used as the training data for the probing tasks. (3) We freeze the pre-trained encoder and obtain span representations using token representations from layer L_i. (4) We train multi-layer perceptrons (MLPs) for each of the probing tasks using pseudo-labels. (5) We fine-tune the pre-trained encoder using sentence representation of the input to achieve the end tasks. Figure 1 shows the entire end-to-end model probing framework.

We used nine different text classification tasks from the GLUE benchmark (Wang et al., 2018) as end tasks and four tasks that cover syntactic and semantic properties (Dependency Parsing, Part-of-Speech Tagging, Named Entity Recognition, and Semantic Role Labeling) as probing tasks.

3.2.1 Probing Tasks

Part-of-Speech Tagging (PoS) is the process of tagging a specific part-of-speech label for each token in sentence. We take a span with length 1 containing token representation h_i at position i as probing input and predict the corresponding PoS tag using the Flair PoS tagger from HuggingFace.

Named Entity Recognition (NER) classifies entity mentions in unstructured text into pre-defined entity categories for a single span. We use the NER tagger from Flair with 4 entities: person names (PER), organizations (ORG), locations (LOC), and miscellaneous (MISC).

Dependency Parsing (DEP) examines the dependencies between the phrases of a sentence in

order to determine its grammatical structure. We take the span representation s_i and s_j as input and predict the dependency relation between the tokens i and j. We use the DEP tagger from StanfordNLP with 50 different grammatical relations.

Semantic Role Labeling (SRL) is a semantic analysis technique that analyzes the predicate-argument structure of sentences. It focuses on the predicate of the sentence and aims to predict the relationship between the components in the sentence and the predicate. We use the SRL tagger from AllenNLP with 64 labels.

3.2.2 Probing Classifier

A standard probing classifier trains the probing task using internal representations from the target model with target model weights frozen. Instead, we use edge probing to obtain token representations of a span from the pretrained model and train a classifier (multi-layer perceptron) to predict a given linguistic property (probing task). Specifically, we convert all language property prediction tasks into span classification tasks and the representations from start to end tokens are concatenated as span representation inputs to probing classifiers. This allows the performance of the probing classifier to reflect whether the pretrained encoder has learned relevant information for the property.

3.2.3 Models

In order to investigate the quality of language representations, we conduct probing tasks on two transformer-based pretrained encoders, BERT and RoBERTa.

BERT Devlin et al. is designed to pre-train deep bidirectional representations from an unlabeled text by jointly conditioning on both left and right context in all layers. It has become the baseline for most natural language processing tasks due to its conceptual simplicity and empirical success.

RoBERTa Liu et al. combines the removing the Next Sentence Prediction (NSP) objective, training with larger batches and longer sequences, and dynamically changing the mask pattern to achieve a better performance than BERT.

3.2.4 Evaluation

For the end-task evaluation, we use Matthews Correlation for CoLA, Pearson Correlation for STSb, and accuracy for the rest of the GLUE benchmark tasks. We use accuracy for all four of the probing tasks. Since we may have multiple pseudo-labels for a single instance, we aggregate accuracies of probing predictions in one instance to an instance-level metric, then aggregate all instance-level metrics in a dataset to a system-level metric.

4 Experiments

4.1 Experiment Settings

We probe all layers by extracting representations from *bert-base-uncased* and *robert-base*, hence having two pre-trained models, 12 layers for each pre-trained model, 4 probing tasks, and 9 end tasks. This results in 216 systems, for which we identify each system by $\{endtask_i\}\_\{model_j\}\_\{layer_k\}$, e.g. *bert-base-uncased_cola_12*.

4.2 System-Level Analysis

In this section, we present system-level probing results under end-to-end settings. We generate pseudo-labels for each input sentence of end tasks and probed multiple language models with the same input sentences on both probing and end tasks.

Previous works have found that lower layers of a language model encode more local syntax while higher layers capture more complex semantics (Van Aken et al., 2019). In addition to simply obtaining the overall results for each model, we also aim to study how information on linguistic properties is encoded in different layers of the pre-trained encoder.

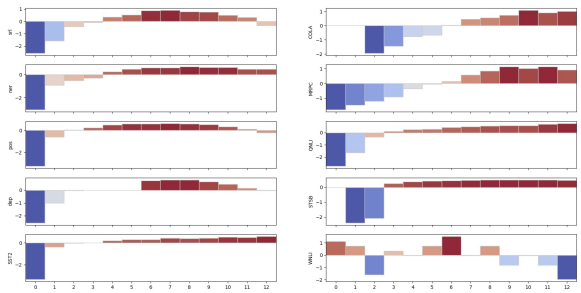

Figure 2: Layer-wise probing performance of *bert-base-uncased* model after Z-normalization. Higher values indicate better performance.

Figure 2 shows the z-normalized performance on each individual task. We noticed that probing tasks perform better with representations of middle layers. For semantics-related end tasks, the models with more layers outperform more shallow models, which indicates that global semantic information from higher layers plays a crucial role in solving

complex tasks like natural language inference and paraphrase detection.

4.2.1 Probing Task Correlation Analysis

In addition to understanding layer-wise behavior, probing results also shed light on relationships between different NLP tasks as shown in Figure 3. We further analyzed the pair-wise statistical correlation between tasks based on the probing score of several candidate models. In the correlation heatmap, we have conducted end-to-end probing on a candidate model m_j, thus obtaining the performance score on the i-th end task $s_{i,j}$. Accordingly, we also evaluated the model on four probing datasets generated by SOTA taggers and extracted the probing performances $\{s_{i,j}^{ner}, s_{i,j}^{pos}, s_{i,j}^{srl}, s_{i,j}^{dep}\}$. All experiment results are aggregated across different models to get performance vectors for the i-th end task $\{\mathbf{s}_i, \mathbf{s}_i^{ner}, \mathbf{s}_i^{pos}, \mathbf{s}_i^{srl}, \mathbf{s}_i^{dep}\}$. We further calculated the Pearson's correlation coefficients between all pairs of end and probing task performance vectors.

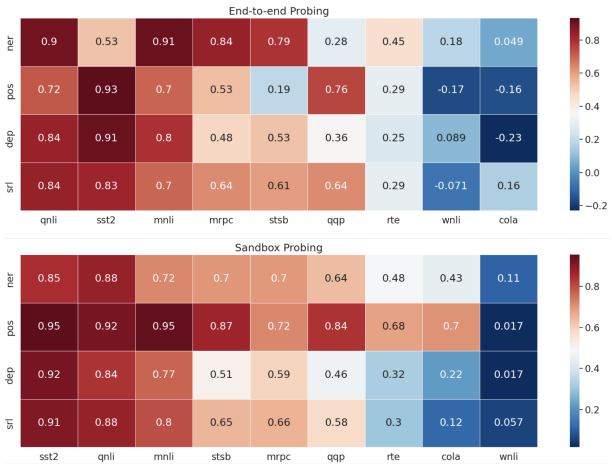

Figure 3: Pearson correlation score matrix under sandbox(up) and end-to-end(bottom) probing settings.

In Figure 3, we observed several probing tasks showing high correlations with end tasks. Firstly, Named Entity Recognition and Dependency Parsing showed a consistently high correlation with natural language inference tasks (QNLI and MNLI), which indicates that models with a better understanding of the sentence structure and proper nouns can better identify the entailment relations between two sentences. SRL and POS tagging show a stronger correlation with the QQP task than others, indicating that successfully detecting the semantic relationships between the predicates and arguments is crucial in paraphrase detection.

We also notice that CoLA and WNLI have consistently low correlations under both sandbox and end-to-end settings. As for CoLA, The input sentences are very short and the syntactic structures tend to be simple. Both shallow and deep models can easily achieve high performance on probing tasks, while only deep models can achieve a decent performance for the end task. For WNLI, the end task itself is quite difficult, hence all candidate models' performances are more or less the same, which leads to low correlation.

The above observations dovetail with our hypotheses regarding the desired linguistic properties for the final task. As more probing experiments are performed on a greater number of heterogeneous language models, a more comprehensive set of correlation coefficient results will be collected.

4.2.2 System-Level Ablation Study with Regression Model

We perform a regression analysis to predict the end task performance using the probing task performance as input features. Our baseline is trained with random values sampled from Gaussian distribution with a mean of 0 and a standard deviation of 0.1 [2]. To evaluate the regression model, we use the relative reduction rate of the Root Mean Square Error (RMSE) compared to the baseline model. We obtain all RMSE with 5-fold cross validation and conduct ablation study by removing one probing task at a time. Table 1 shows that different end tasks acquire different probing task information, and using performance of all probing tasks does not result in the highest prediction performance. We notice that the English characters in CoLA only has lower case, and thus the probing task taggers find it difficult to obtain high quality pseudo-labels.

4.3 Fine-Grained Probing Analysis

In the previous section, we discussed how to use holistic performance scores to predict end-task performance and calculate task correlations. Another advantage of end-to-end probing over sandbox probing is that we are able to conduct a more fine-grained instance-level probing. The conventional out-of-domain sandbox probing methods can only analyze at the system level due to the misalignment of probing data and end-task data. However, as is shown in Figure 1, end-to-end probing generates aligned probing performance for each end-task

[2]Monte Carlo simulations with N=500

	CoLA	SST	STSB	WNLI	QNLI	MNLI	RTE	QQP	MRPC
-POS	17.39%	34.95%	49.27%	6.46%	**50.22%**	**51.05%**	39.75%	43.85%	56.28%
-NER	12.10%	42.05%	36.89%	**12.13%**	25.80%	34.92%	45.00%	34.79%	32.34%
-DEP	**23.93%**	**57.69%**	49.45%	1.60%	48.39%	54.48%	54.65%	**47.12%**	48.60%
-SRL	22.08%	56.01%	**52.58%**	6.44%	48.17%	50.88%	**61.32%**	31.49%	**57.56%**
All	14.11%	44.29%	52.32%	10.76%	43.79%	50.00%	55.71%	30.24%	55.78%

Table 1: System-level ablation study to predict end-task performance with probing task features. The metric represent relative reduction rate of RMSE comparing to baseline model. -POS means removing POS performance from the feature set.

	POS	NER	DEP	SRL	CoLA	SST	MRPC	WNLI	QNLI	RTE	QQP
baseline					0.685	0.519	0.663	0.560	0.500	0.550	0.819
+POS	✔				0.805	0.923	0.854	0.560	0.908	0.568	0.903
+NER		✔			0.852	0.907	0.848	0.644	0.905	0.627	0.925
+DEP			✔		0.800	0.921	0.854	0.547	0.910	0.561	0.910
+SRL				✔	0.803	0.902	0.839	0.520	0.908	0.619	0.909
Full model	✔	✔	✔	✔	0.830	0.917	0.868	0.622	0.912	0.694	0.912

Table 2: Instance-level ablation study to predict end-task performance with probing task features.

instance, which makes it possible for fine-grained error analysis and model evaluation.

4.3.1 Case Error Analysis with Probing Results

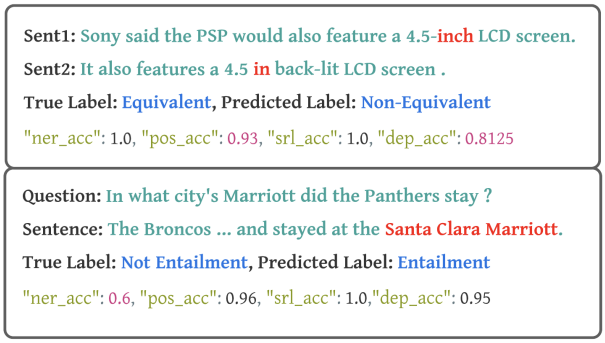

Figure 4: Error cases sampled and adapted from MRPC (up) and QNLI (bottom) development split. The instance-level accuracies correspond to the probing results of the second sentence in each case.

We first introduce a use case for end-to-end probing on fine-grained error analysis. In Figure 4, we select two failed cases in MRPC and QNLI to analyze their causes of failure. The first case is paraphrase classification, which is to determine whether a pair of sentences have the same meaning. We can see that the instance-level POS and DEP scores are low. After parsing the probing predictions, we found that the model misclassified the word *in* as a preposition instead of a noun, which thus led to the incorrect prediction of the dependency tree structure. This could be the reason why the probed model thinks the sentence pairs are not equivalent. The second case is sampled from the task QNLI, which aims to test whether a sentence contains the answer to a given question. The question is asking about the Panthers, and the sentence is describing the Broncos, so the sentence doesn't have a relevant answer, but the model thinks it does. One might simply attribute the failure to the model's lack of long-range context dependencies, yet we also noticed that the model misclassified *Santa Clara Marriott* as an organization instead of a location in the probing test, which indicates that the model also failed on local context understanding.

4.3.2 Instance-Level Ablation Study

To better understand which linguistic properties a particular end task relies on, we built a classification model which takes probing performances as features to predict the true label for the end task. As discussed in section 4.2.1, we aggregate instance-level probing results for each model on the in-domain probing datasets, then take the average of the performance scores across all the tested models as the training features. We add one feature at a time to see the performance change in the prediction accuracy. We choose a trivial baseline model, which only takes the frequency of model predictions as the single feature, which is equivalent to the majority vote. If the prediction score changes a lot after ablation, this linguistic feature

has a greater impact on the final task.

The results show that after adding instance-level probing performance, the prediction accuracy increases from 0.2 to 0.4. The POS feature plays an important role in sentiment analysis. Dependency parsing shows more influence on datasets with longer context and complex sentence structure, such as MRPC and QNLI.

5 Related Works

5.1 Model Performance Prediction

Recent work has consistently demonstrated the validity of predicting model performance without training the model. (Xia et al., 2020) used a collection of language typological and statistical features on machine translation and cross-lingual NLP tasks. (Ye et al., 2021) breaks down the holistic performance into different interpretable parts for fine-grained performance prediction. Instead of generating features from datasets, (Zhu et al.) conducted probing tests on several out-of-domain diagnose datasets and utilized the probing scores to predict the finetune performance. In end-to-end probing, we also use the probing scores as the prediction features. However, the main difference is that we investigated both the in-domain and out-domain probing datasets.

6 Conclusion

We introduce an end-to-end probing framework that resolves the disconnect between the probing and end tasks and have confirmed all three of the hypotheses to be true: (1) nontrivial correlations between probing tasks and end tasks exist (2) different end tasks target different linguistic abilities (3) end-to-end probing can provide fine-grained insights. We anticipate a more comprehensive analysis to be made available as we include additional natural language inference tasks.

Our end-to-end model probing framework is model agnostic and can be applied to other end tasks, such as text generation tasks, and to other domains such as bioinformatics. Researchers can gain insights into the inner workings of the model on a variety of tasks and domains by analyzing strengths and weaknesses through system-level and instance-level performance analysis. The future research direction can focus on developing a strategy to improve the model on its weaknesses and reiterate the end-to-end model probing process.

References

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for NLP. *CoRR*, abs/1712.06751.

Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M Drucker. 2019. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13.

Qiao Jin, Bhuwan Dhingra, William W Cohen, and Xinghua Lu. 2019. Probing biomedical embeddings from language models. *arXiv preprint arXiv:1904.02181*.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaicheng Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, Zi-Yi Dou, and Graham Neubig. 2021. Explainaboard: An explainable leaderboard for nlp. *arXiv preprint arXiv:2104.06387*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nedjma Djouhra Ousidhoum, Xinran Zhao, Tianqing Fang, Yangqiu Song, and Dit Yan Yeung. 2021. Probing toxic content in large pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. 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. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Betty Van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. 2019. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1823–1832.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8625–8646.

Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. 2017. Robust multilingual part-of-speech tagging via adversarial training. *arXiv preprint arXiv:1711.04903*.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable nlp performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714.

Fan Yin, Quanyu Long, Tao Meng, and Kai-Wei Chang. 2020. On the robustness of language encoders against grammatical errors. *CoRR*, abs/2005.05683.

Zining Zhu, Soroosh Shahtalebi, and Frank Rudzicz. Predicting fine-tuning performance with probing. In *Challenges* {\&, year=2022.