Probing the Probing Paradigm:
Does Probing Accuracy Entail Task Relevance?

Abhilasha Ravichander[1] **Yonatan Belinkov**[2*] **Eduard Hovy**[1]
[1]Language Technologies Institute, Carnegie Mellon University
[2]Technion – Israel Institute of Technology
aravicha@cs.cmu.edu
belinkov@technion.ac.il, hovy@cmu.edu

Abstract

Although neural models have achieved impressive results on several NLP benchmarks, little is understood about the mechanisms they use to perform language tasks. Thus, much recent attention has been devoted to analyzing the sentence representations learned by neural encoders, through the lens of 'probing' tasks. However, to what extent was the information encoded in sentence representations, as discovered through a probe, actually used by the model to perform its task? In this work, we examine this probing paradigm through a case study in Natural Language Inference, showing that models can learn to encode linguistic properties even if they are not needed for the task on which the model was trained. We further identify that pretrained word embeddings play a considerable role in encoding these properties rather than the training task itself, highlighting the importance of careful controls when designing probing experiments. Finally, through a set of controlled synthetic tasks, we demonstrate models can encode these properties considerably above chance-level *even when distributed in the data as random noise*, calling into question the interpretation of absolute claims on probing tasks.[1]

1 Introduction

Neural models have established state-of-the-art performance on several NLP benchmarks (Kim, 2014; Seo et al., 2017; Chen et al., 2017; Devlin et al., 2019). However, these models can be opaque and difficult to interpret, posing barriers to widespread adoption and deployment in safety-critical or user-facing settings (Belinkov and Glass, 2019). How can we know what information, if any, neural models learn and leverage to perform a task? This question has spurred considerable community effort to develop methods to analyze neural models, motivated by interest not just to have models perform tasks well, but also to understand the mechanisms by which they operate.

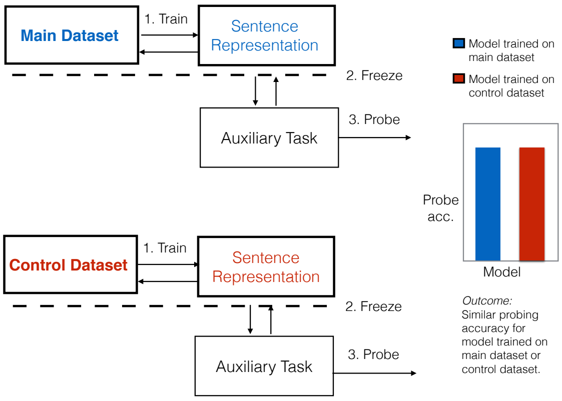


Figure 1: Illustration of our control dataset methodology for evaluating probing classifiers. Control datasets are constructed such that a linguistic feature is not discriminative with respect to the task. Representations from models trained on the main dataset and control dataset are probed for the linguistic feature, and demonstrate similiar probing performance.

A popular approach to model introspection is to associate the representations learned by the neural network with linguistic properties of interest, and examine the extent to which these properties can be recovered from the representation (Adi et al., 2017). This paradigm has alternatively been called probing (Conneau et al., 2018), auxiliary prediction tasks (Adi et al., 2017) and diagnostic classification (Veldhoen et al., 2016; Hupkes et al., 2018). As an example of this approach, let us walk through an application to analyze information about tense stored in a Natural Language Inference (NLI) model. In Conneau et al. (2018), three sentence-encoder models are trained on a

[*] Supported by the Viterbi Fellowship in the Center for Computer Engineering at the Technion.

[1] Code and data available at https://github.com/AbhilashaRavichander/probing-probing.

NLI dataset (MultiNLI; Williams et al., 2018). The encoder weights are frozen, and the encoders are then used to form sentence representations for the auxiliary task— predicting the tense of the verb in the main clause of the sentence. A separate classifier, henceforth called the probing classifier, is trained to predict this property based on the constructed representation. The probing task itself is typically selected to be relevant to the training task, and high probing performance is considered as evidence that the property is encoded in the learned representation. Due to its simplicity, a growing body of work uses this approach to pinpoint the information models rely on to do a task (Alt et al., 2020; Giulianelli et al., 2018; Saleh et al., 2020).

In this work, we examine the connection between the information encoded in a representation and the information a model relies on. Through a set of carefully designed experiments on the benchmark SentEval probing framework (Conneau et al., 2018), we shed light on information use in neural models. Our story unfolds in four parts:

1. First, we establish careful control versions of the training task such that task performance is invariant to a chosen linguistic property (Figure 1). We show that even when models cannot use a linguistic property to perform the task, the property can be reliably recovered from the neural representations through probing (§4.1).

2. Word embeddings could be a natural suspect for this discrepancy. We demonstrate that initializing models with pretrained word embeddings does play a role in encoding some linguistic properties in sentence representations. We speculate that probing experiments with pretrained word embeddings conflate two tasks — training word embeddings and the main task under consideration (§4.2).

3. What happens if we neutralize the effect of pre-trained word embeddings? Even when word embeddings are trained from scratch, we demonstrate that models still encode linguistic properties when they are not actually required for a task (§4.3).

4. Finally, through a carefully controlled synthetic scenario we demonstrate that neural models can encode information incidentally, even if it is distributed as random noise with respect to the training task (§5). We discuss several considerations when interpreting the results of probing experiments and highlight avenues for future research needed in this important area of understanding models, tasks and datasets (§6).

2 Background and Related Work

Progress in Natural Language Understanding (NLU) has been driven by a history of defining tasks and corresponding benchmarks for the community (Marcus et al., 1993; Dagan et al., 2006; Rajpurkar et al., 2016). These tasks are often tied to specific practical applications, or to developing models demonstrating competencies that transfer across applications. The corresponding benchmark datasets are utilized as proxies for the tasks themselves. How can we estimate their quality as proxies? While annotation artifacts are one facet that affects proxy-quality (Gururangan et al., 2018; Poliak et al., 2018; Kaushik and Lipton, 2018; Naik et al., 2018; Glockner et al., 2018), a dataset might simply not have coverage across competencies required for a task. Additionally, it might consist of alternate "explanations", features correlated with the task label in the dataset while not being task-relevant, which models can exploit to give the impression of good performance at the task itself.

Two analysis methods have emerged to address this limitation: 1) **Diagnostic examples**, where a small number of samples in a test set are annotated with linguistic phenomena of interest, and task accuracy is reported on these samples (Williams et al., 2018; Joshi et al., 2020). However, it is difficult to determine if models perform well on diagnostic examples because they actually learn the linguistic competency, or if they exploit spurious correlations in the data (McCoy et al., 2019; Gururangan et al., 2018; Poliak et al., 2018). 2) **External challenge tests** (Naik et al., 2018; Isabelle et al., 2017; Glockner et al., 2018; Ravichander et al., 2019; McCoy et al., 2019), where examples are constructed, either through automatic methods or by experts, exercising a specific phenomenon in isolation. However, it is challenging and expensive to build these evaluations, and non-trivial to isolate phenomena (Liu et al., 2019).

Thus, **probing** or *diagnostic classification* presents a compelling alternative, wherein learned representations can directly be probed for linguistic properties of interest (Ettinger et al., 2016; Belinkov et al., 2017; Adi et al., 2017; Tenney et al., 2019; Zhang and Bowman, 2018; Warstadt et al., 2019). There has been a variety of research that employs probing to test hypotheses about the mechanisms models used to perform tasks. Shi et al. (2016) examine learned representations in machine translation for syntactic knowledge. Vanmassenhove et al. (2017) investigate aspect in neural machine translation systems, finding that tense information could be extracted from the encoder, but that part of this information may be lost when decoding. Conneau et al. (2018) use probing to examine the correlation between linguistic properties and downstream tasks (including MT and NLI). Hupkes et al. (2018) train a 'diagnostic classifier' to extract information from a sequence of hidden representations in a neural network. If the classifier achieves high accuracy, it is concluded that the network is keeping track of the hypothesized information. Giulianelli et al. (2018) use diagnostic classifiers to predict number from the internal states of a language model. Kim et al. (2019) study what different NLP tasks teach models about function word comprehension. Alt et al. (2020) analyze learned representations in relation extraction, through a set of fourteen probing tasks for relevant linguistic properties. Saleh et al. (2020) examine the representations learned by neural dialog models for insights into what the model learns about engaging in dialog. See the survey by Belinkov and Glass (2019) for many more examples.

Closely related to our work is that of Hewitt and Liang (2019), which studies the role of lexical memorization in probing, and recently the work of Pimentel et al. (2020) and Voita and Titov (2020) who analyze probing from an information-theoretic perspective. These works join an ongoing debate on the correct way to characterize the expressivity of the probing classifier, with the latter proposing ease of extractability as a criterion for selecting appropriate probes. Our work pursues an orthogonal line of inquiry, demonstrating that relying on diagnostic classifiers to interpret model reasoning for a task suffers from a fundamental limitation: properties may be incidentally encoded even when not required for a task. Thus, our work is also related to a broader investigation of how neural models encode information (Tishby and Zaslavsky, 2015; Voita et al., 2019), studying to what extent information encoded in neural representations is indicative of information needed to perform tasks.

3 Methodology

In this section we describe our modified probing pipeline (Figure 1), where we construct control datasets, such that a particular linguistic feature is not required in making task judgements.[2] Control datasets are based on the intuition that a linguistic feature is not informative for a model to discriminate between classes if the linguistic feature remains constant across classes. For a task label T and linguistic property L, when every example in the control dataset has the same value for L, the linguistic property L in isolation is not discriminative of the task label .

To construct control datasets we hold constant the relevant property value across the whole dataset. In practice, the control datasets are constructed from existing large-scale datasets by partitioning them on the value of a linguistic property, such that every example in the sampled dataset has the same value of linguistic property.[3] They are designed with the following considerations:

1. The linguistic property of interest is auxiliary to the main task and a function of the input, but not of the task decision.
2. Every sample in the training and test sets has the same fixed value of the linguistic property.
3. The training set is large in order to train parameter-rich neural classifiers for the task.

We next describe our main training task, our three auxiliary prediction tasks, and procedures to construct control datasets corresponding to each auxiliary property. Models are trained either on datasets constructed for the main task, or on control datasets, and then probed for the auxiliary property using data from a probing dataset. In this work, we use the experimental settings of Conneau et al. (2018) for both the training task and probing task, due to its popularity as a probing benchmark. However, the conclusions we draw are meant to illustrate the limits and generality of probing as a diagnostic method, rather than discuss the specific experimental settings of Conneau et al. (2018).

Main Task: We study the Natural Language Inference (NLI) training task from Conneau et al. (2018) as the main task for training sentence en-

[2] While our motivating example of a task is natural language inference, we expect control datasets can be constructed for most text classification tasks with a small finite label space.

[3] All probing tasks in this work take a sentence representation as input and perform mappings to binary labels $\{0, 1\}$.

Linguistic Control	Property	# Train	# Test
MultiNLI	-	392,702	20,000
Tense	Past	69,652	1678
Subject Number	Singular	102,452	2584
Object Number	Singular	43,178	1060

Table 1: Statistics of control datasets partitioned by linguistic property.

coders. NLI is a benchmark task for research on natural language understanding (Cooper et al., 1996; Haghighi et al., 2005; Harabagiu and Hickl, 2006; Dagan et al., 2006; Giampiccolo et al., 2007; Zanzotto et al., 2006; MacCartney, 2009; Dagan et al., 2010; Marelli et al., 2014). Broadly, the goal of the task is to decide if a given hypothesis can be inferred from a premise in a justifiable manner. Typically, this is framed as the 3-way decision of whether a hypothesis is true given the premise (entailment), false given the premise (contradiction), or whether the truth value cannot be determined (neutral). We use MultiNLI (Williams et al., 2018), a broad-coverage NLI dataset, to train sentence encoders.

Auxiliary Tasks: We consider three tasks that probe sentence representations for semantic information from Conneau et al. (2018), all of which "require some understanding of what the sentence denotes". We construct the probing datasets such that lexical items that are associated with the probing task do not occur across the train/dev/test split for the target. This design controls for the effect of memorizing word types associated with target categories (Hewitt and Liang, 2019). The tasks considered in this study are:

1. Tense: Categorize sentences based on the tense of the main verb.
2. Subject Number: Categorize sentences based on the number of the subject of the main clause.
3. Object Number: Categorize sentences based on number of the direct object of the main clause.

Control: For each auxiliary task, we partition MultiNLI such that premises and hypotheses agree on a single value of the linguistic property. For example, for the auxiliary task Tense, sentences with VBP/VBZ/VBG forms are labeled as present and VBD/VBN as past tense.[4] Subsequently, premise-hypothesis pairs where the main verbs in both premise and hypothesis are in past tense are extracted from train/dev sets to form the control datasets for tense. Thus, every sentence in the dataset (both premises and hypotheses), has the same value of the auxiliary property.[5]

This procedure results in three control datasets/tasks: MultiNLI-PastTense, MultiNLI-SingularSubject, and MultiNLI-SingularObject. For all three, we fix the value of the linguistic property to the one that results in the maximum number of training instances on partitioning, namely fixing past tense, singular subject number, and singular object number. Descriptive statistics for each dataset appears in Table 1.

Models: We use CBOW and BiLSTM-based sentence-encoder architectures. The choice of these models is motivated by their demonstrated utility as NLI architectures (Williams et al., 2018), and because their learned representations have been extensively studied for the three linguistic properties used in this work (Conneau et al., 2017).[6]

1. Majority: The hypothetical performance of a classifier that always predicts the most frequent label in the test set.

2. CBOW: A simple Continuous Bag-Of-Words Model (CBOW). The sentence representation is the sum of word embeddings of constituent words. Word embeddings are finetuned during training.

3. BiLSTM-Last/Avg/Max: For a sequence of N words in a sentence $s = w_1 ... w_n$, the bidirectional LSTM (BiLSTM; Hochreiter and Schmidhuber (1997)) computes N vectors extracted from its hidden states $\vec{h}_1, ..., \vec{h}_n$. We produce fixed-length vector representations in three ways: by selecting the last hidden state h_n (BiLSTM-Last), by averaging the produced hidden states (BiLSTM-Avg) or by selecting the maximum value for each dimension in the hidden units (BiLSTM-Max).

[4] These heuristics are specific to English, as is MultiNLI. We use the Stanford Parser for constituency, POS and dependency parsing (Manning et al., 2014).

[5] This procedure replicates the original SentEval probing labels (Conneau et al., 2018) with 89.37% accuracy on tense, 87.77% accuracy on subject number and 88.19% accuracy on object number.

[6] We leave an exploration of recent transformer-based architectures to future work, noting however that this study stands alone as evidence that probing performance does not correlate to task importance.

	Tense		Subject Number		Object Number	
	Dev-ST	Probing	Dev-SS	Probing	Dev-SO	Probing
Majority	37.90	50.00	36.88	50.0	39.52	50.0
CBOW-DS	57.57	82.36	58.4	76.55	55.85	75.49
CBOW-PT	60.31	82.2	58.2	75.69	59.15	74.38
BiLSTM-Av-DS	63.53	82.93	64.24	79.53	66.23	76.11
BiLSTM-Av-PT	65.08	82.79	66.76	78.81	67.08	75.48
BiLSTM-Max-DS	63.35	81.14	65.91	78.56	65.94	74.79
BiLSTM-Max-PT	64.6	81.04	66.87	79.51	66.98	72.44
BiLSTM-Last-DS	61.08	80.43	64.2	81.52	62.26	72.65
BiLSTM-Last-PT	63.89	78.44	66.18	78.9	66.04	72.82

Table 2: Performance comparisons of task-controlled (PT) and downsampled models (DS). Dev-ST, Dev-SS and Dev-SO is the MultiNLI development set controlled for tense, subject number and object number, respectively. PT is a model trained on data partitioned by linguistic property—these models should not be able to leverage the linguistic property to perform their training task. DS is models trained on downsampled MNLI data to match the number of instances in partitioned. Majority baseline reflects distribution of main task classes for controlled development sets (Dev-ST, Dev-SS and Dev-SO), or class distribution of auxiliary property for probing datasets. We can observe that models consistently display similar probing accuracies whether the property was needed for the training task or not (Probing). Competitive performance of PT model variants to DS model variants on controlled MNLI development sets (Dev-ST, Dev-SS, Dev-SO) validates the controlled linguistic property is not useful to solve the controlled version of the task.

All models produce separate sentence vectors for the premise and hypothesis. They are concatenated with their element-wise product and difference (Mou et al., 2016), passed to a `tanh` layer and then to a 3-way softmax classifier. Models are initialized with 300D GloVe embeddings (Pennington et al., 2014) unless specified otherwise, and implemented in Dynet (Neubig et al., 2017). After the model is trained for the NLI task, the learned sentence vectors for the premise and hypothesis are probed. The probing classifier is a 1-layer multi-layered perceptron (MLP) with 200 hidden units.

4 Probing the Probing Paradigm

4.1 Probing with Linguistic Controls

As a first step, we ask the question: to what extent is the information encoded in learned representations, as reflected in probing accuracies, driven by information that is useful for the training task? We construct multiple versions of the task (both training and development sets) where the entailment decision is independent of the given linguistic property, through careful partitioning as described in §3. To control for the effect of training data size, we downsample MultiNLI training data to match the number of samples in each partitioned version of the task. These results are in Table 2.

Strikingly, we observe that even when models are trained on tasks that *do not require the linguistic property at all* for the main task (rows with PT in Table 2), probing classifiers still exhibit high accuracy (sometimes up to ∼80%). Probing data is split lexically by target across partitions, and thus lexical memorization (Hewitt and Liang, 2019) cannot explain why these properties are encoded in the sentence representations. Across models, on the version of the task where a particular linguistic property is not needed, classifiers trained on data that does not require that property perform comparably to classifiers trained on MultiNLI training data (DS vs PT models, on Dev-ST, Dev-SS, and Dev-SO).

4.2 Effect of Word Embeddings

A potential explanation lies in our definition of a "task". Previous work directly probes models trained for a target task such as NLI. However, when models are initialized with pre-trained word embeddings, the conflated results of two tasks are being probed – the main training task of interest, and the task that was used to train the word embeddings. Both tasks may contribute to the encoding of information in the learned representation, and it is unclear to what extent they interact. Previous work has noted the considerable amount of information

	Tense		Subject Number		Object Number	
	Dev	Probing	Dev	Probing	Dev	Probing
Majority	36.50	50.0	36.50	50.0	36.50	50.0
CBOW-Word	62.21	83.74	62.1	76.91	61.93	75.4
CBOW-Rand	56.98	60.14	56.27	67.01	56.82	64.71
BiLSTM-Avg-Word	70.05	82.48	70.67	76.53	69.82	72.29
BiLSTM-Avg-Rand	63.33	61.4	64.0	67.68	63.71	63.87
BiLSTM-Max-Word	68.67	78.34	69.19	73.96	69.12	68.53
BiLSTM-Max-Rand	62.78	62.89	63.29	69.51	63.28	62.84
BiLSTM-Last-Word	68.32	74.61	69.04	71.82	68.82	69.27
BiLSTM-Last-Rand	62.14	62.96	61.88	67.45	62.29	61.32

Table 3: Performance comparisons of models initialized with pretrained word embeddings (Word) and models with randomly initialized embeddings (Rand) on MNLI Development Set (Dev) and on the probing task (Probing). Embeddings are updated during task-specific training. We can observe that probing performance decreases sharply for all models when word embeddings are randomly initialized, suggesting a considerable component of probing performance comes from pretraining word embeddings rather than what a model learns during the task.

present in word embeddings, and proposed methods to measure this effect, such as comparing with bag-of-word baselines or random encoders (Wieting and Kiela, 2018). However, these methods fail to isolate the contribution of the training task.

To study this, we compare models initialized with pre-trained word embeddings (Pennington et al., 2014) and then trained for the main task, to models initialized with random word embeddings and then updated during the main task. These results are presented in Table 3. We observe that probing accuracies drop across linguistic properties in this setting (compare rows with Word and Rand in the table), indicating that models with randomly initialized embeddings generate representations that contain less linguistic information than the models with pretrained embeddings. This result calls into question how to interpret the contribution of the main task to the encoding of a linguistic property, when the representation has already been initialized with pre-trained word embeddings. The word embeddings could themselves encode a significant amount of linguistic information, or the main task might contribute to encoding information in a way already largely captured by word embeddings.

4.3 How do models encode linguistic properties?

When we isolate the effect of the main task with randomly initialized word embeddings, are properties not predictive of the main task judgement *still* being encoded? To study this, we revisit our linguistic control tasks but train all models with randomly initialized word embeddings. We also train comparable models on downsampled MultiNLI training data. These results can be found in Table 4. We observe that even in the setting with randomly initialized word embeddings, these properties are still encoded to a similar extent (and above the majority baseline) in the downsampled and control versions of their task.

5 A Synthetic Experiment: Analyzing Encoding Dynamics

We have demonstrated that models encode properties even when they are not required for the main task. Thus, probing accuracy cannot be considered indicative of competencies any given model relies on. What circumstances could lead to models encoding properties incidentally? Can we determine when a linguistic property is not needed by a model for a task? To study this, we build carefully controlled synthetic tests, each capturing a kind of noise that could arise in datasets.

5.1 Synthetic Task

We consider a task where the premise P and hypothesis H are strings from $S = \{(a|b)(a|b|c)^*\}$ of maximum length 30, and the hypothesis H is said to be entailed by the premise P if it begins with the same letter a or b,[7] for example:

[7] A task with a similar objective was used by Belinkov et al. (2019a) to demonstrate unlearning bias in datasets. The task is equivalent to XOR, which is learnable by an MLP.

	Tense		Subject Number		Object Number	
	Dev-ST	Probing	Dev-SS	Probing	Dev-SO	Probing
Majority	37.90	50.0	36.88	50.0	39.52	50.0
CBOW-Rand-DS	49.88	61.33	51.04	67.32	49.25	63.63
CBOW-Rand-PT	53.28	61.37	50.97	67.02	52.45	63.84
BiLSTM-Avg-Rand-DS	57.21	63.75	60.76	68.5	59.53	63.89
BiLSTM-Avg-Rand-PT	60.91	63.07	61.18	69.12	60.57	63.77
BiLSTM-Max-Rand-DS	59.18	61.05	61.8	70.32	60.57	64.68
BiLSTM-Max-Rand-PT	60.55	61.53	63.78	70.6	63.49	64.26
BiLSTM-Last-Rand-DS	56.73	63.88	58.82	69.09	56.79	63.86
BiLSTM-Last-Rand-PT	57.39	62.88	61.88	68.8	60.75	61.96

Table 4: Performance of task-controlled (PT) and downsampled models (DS), when word embeddings are trained from scratch. (Rand) indicates the model is initialized with random embeddings, rather than pretrained embeddings. Dev-ST, Dev-SS and Dev-SO is the MultiNLI development set controlled for tense, subject number and object number, respectively. We observe that when the training task is isolated in this way, for all models probing performance is similar whether a linguistic property is necessary for the task or not (Probing).

(a, ab) → Entailed	(a, ba) → Not Entailed
(b, ba) → Entailed	(b, ab) → Not Entailed
(b, bc) → Entailed	(b, acb) → Not Entailed

Consider the auxiliary task of predicting whether a sentence contains the character c from a representation, analogous to probing for a task-irrelevant property. We sample premises/hypotheses from a set of strings $S' = (a|b)^*$ of maximum length 30, and simulate four kinds of correlations that could occur in a dataset by inserting c at a random position in the string after the first character:[8]

1. NOISE : The property could be distributed as noise in the training data. To simulate this, we insert c into 50% of randomly sampled premise and hypothesis strings.
2. UNCORRELATED : The property could be unrelated to the task decision, but correlated to some other property in the data. To simulate this, we insert c to premises beginning with a.
3. PARTIAL: The property could provide a partial explanation for the main task decision. To simulate this, we insert c to premise and hypothesis strings beginning with a.[9]

Dataset	# Train	# Dev	# Test
NOISE	20000	5000	5000
UNCORRELATED	20000	5000	5000
PARTIAL	20000	5000	5000
FULL	20000	5000	5000
PROBE	23732	5000	5000

Table 6: Descriptive statistics for NOISE, UNCORRELATED, PARTIAL and FULL synthetic datasets, as well as the dataset used to train the probing classifier(PROBE). We ensure that datasets do not have any data leakage in the form of strings appearing across train/dev/test splits, or across task and probing splits in either the main task or the probing dataset.

4. FULL: The property provides a complete alternate explanation for the main task decision. We insert c to premise and hypothesis strings whenever the hypothesis is entailed.

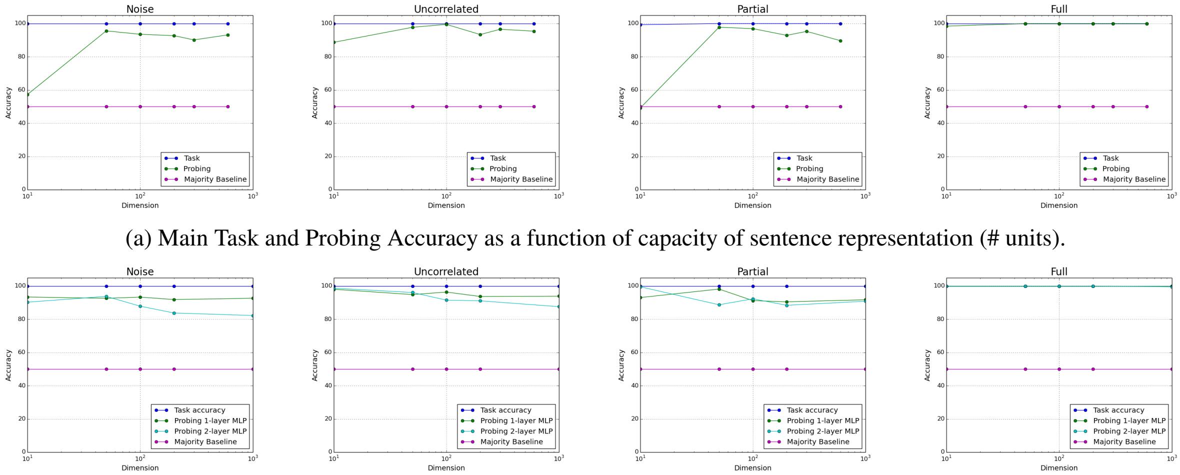
Descriptive statistics of all datasets are in Table 6.

5.2 Results

Figure 2a presents the performance of the model and the probe on the four test sets. We observe that we are able to train a classifier to predict the presence of c considerably above chance-level in all four cases. This is notable, considering that even when the property is distributed as random noise (NOISE) uncorrelated with the actual task, the

[8] We additionally explore the utility of adversarial learning, as a potential approach to identifying properties required by a model to perform a task, by suppressing a property and measuring task performance (Appendix. A). We find in our exploration that adversarial approaches are not completely successful at suppressing the linguistic property under consideration, though capacity of the adversary could play a role.

[9] Models can use either the presence of c, or the first character of the strings being a to make their prediction, but they must use whether the first character of the strings is b.



(a) Main Task and Probing Accuracy as a function of capacity of sentence representation (# units).

(b) Main Task and Probing Accuracy as a function of capacity of probing classifier (# units).

Figure 2: Task and probing accuracy of (BiLSTM, last) on Noise, Uncorrelated, Partial and Full synthetic datasets.

model encodes it. This simple synthetic task suggests that models learn to encode linguistic properties *incidentally*, implying it is a mistake to rely on the accuracy of probes to measure what information the model relies upon to solve a task. We further discuss the role of representation capacity and probing classifier expressivity:

Representation size: Lower-capacity models may encode task-specific information at the expense of irrelevant properties. To examine this, we train the BiLSTM architecture with hidden size 10, 50, 100, 200, 300 and 600 units, and train the probing classifier on the auxiliary task. These results are reported in Figure 2a. We observe that while the main task accuracy remains consistent across choice of dimension, probing accuracy decreases for models with lower capacity across categories. This suggests that the capacity of the representation may play a role in which information it encodes, with lower capacity models being less prone to incidentally encoding irrelevant information.

Probing classifier capacity: We examine whether probing classifier capacity is a factor in the incidental encoding of linguistic properties. A more complex probing classifier may be more effective at extracting linguistic properties from representations. We experiment with probing classifiers utilizing 1-layer and 2-layer MLP's of dimensions {10, 50, 100, 200, 1000}. The results are shown in Figure 2b. We find that a higher-capacity probing classifier does not necessarily imply higher probing accuracy. Further, in all the settings of probing classifier capacity we study, we are able to perform the auxiliary task considerably above chance accuracy, even when the property is distributed as random noise.

6 Discussion

We briefly discuss our findings, with the goal of providing considerations for deciding which inferences can be drawn from a probing study, and highlighting avenues for future research.

Linguistic properties can be incidentally encoded: Probing only indicates that some property correlated with a linguistic property of interest is encoded in the sentence representation — but we speculate that it cannot isolate what that property might be, whether the correlation is meaningful, or how many such properties exist. As shown in the controlled synthetic tests, even if a particular property is not needed for a task, the information can be extracted from the representation with high accuracy. Thus, probing cannot determine if the property is actually needed to do a task, and should not be used to pinpoint the information a model is relying upon. A negative result here can be more meaningful than a positive one. Adversarially suppressing the property may help determine if an alternate explanation is readily available to the model, with an appropriate choice of probing classifier. In this case, if the model maintains task accuracy while suppressing the information, one can conclude the property is not needed by the model for the task, but its failure to do so is not indicative of property importance. Causal alternatives to probing classifiers that intervene in model representations to examine effects on prediction also present another promising direction for future work (Giulianelli et al., 2018; Bau et al., 2018; Vig et al., 2020).

Careful controls and baselines: We emphasize the need for probing work to establish careful controls and baselines when reporting experimental results. When probing accuracy for a linguistic competence is high, it may not be directly attributable to the training task. In this work we identify two confounds: incidental encoding and interaction between training tasks. Perhaps future work will determine causes of incidental encoding and identify further baselines and controls that allow reliable conclusions to be drawn from probing studies.

Lack of gold-standard data of task requirements: While prior work has discussed the different linguistic competencies that might be needed for a task based on the results of probing studies, these claims are inherently hard to reliably quantify given that the exact linguistic competencies, as well as the extent to which they are required, are difficult to isolate for most real-world datasets. Controlled test cases (such as those in §5.1) are effective as basic sanity checks for claims based on diagnostic classification, and provide insight into encoding dynamics in sentence representations.

Datasets are proxies for tasks, and proxies are imperfect reflections: Finally, we speculate that while datasets are used as proxies for tasks, they might not reflect the full complexity of the task. Aside from having dataset-specific idiosyncrasies in the form of unwanted biases and correlations, they might also not require the full range of competencies that we expect models to need to succeed on the task. Future work should refine or move beyond the probing paradigm to carefully identify what the competencies reflected in any dataset are, and how representative they are of overall task requirements.

What probes are good for: This work explores only the implications of probing as a diagnostic tool for pinpointing the information models use to do a task. However, when sentence representations are used subsequently downstream (after being trained on the main task), probing can give insight into what information is encoded in the model (irrespective of how that encoding came to be). Future work could include exploring the connection between information encoded in the representation and whether models successfully learn to use them in downstream tasks.

7 Conclusion

The probing paradigm has evoked considerable interest as a useful tool for model interpretability. In this work, we examine the utility of probing for providing insights into what information models rely on to do tasks, and requirements for tasks themselves. We identify several considerations when probing sentence representations, most strikingly that linguistic properties can be incidentally encoded even when not needed for a main task. This line of questioning highlights several fruitful areas for future research: how to successfully identify the set of linguistic competencies necessary for a dataset, and consequently how well any dataset meets task requirements, how to reliably identify the exact information models rely upon to make predictions, and how to draw connections between information encoded by a model and used by a model downstream.

Acknowledgements

This research was supported in part by grants from the National Science Foundation Secure and Trustworthy Computing program (CNS-1330596, CNS15-13957, CNS-1801316, CNS-1914486) and a DARPA Brandeis grant (FA8750-15-2-0277). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NSF, DARPA, or the US Government. This research was also supported by the ISRAEL SCIENCE FOUNDATION (grant No. 448/20). Y.B. was also supported by the Harvard Mind, Brain, and Behavior Initiative. The authors would like to extend special gratitude to Carolyn Rose and Aakanksha Naik, for insightful discussions related to this work. The authors are also grateful to Yanai Elazar, Lucio Dery, Paul Michel, Shruti Rijhwani and Siddharth Dalmia for reviews while drafting this paper, and to Marco Baroni for answering questions about the SentEval probing tasks.

References

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

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Probing linguistic features of sentence-level representations in neural relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1534–1545, Online. Association for Computational Linguistics.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. 2018. Gan dissection: Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019a. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019b. On adversarial removal of hypothesis-only bias in natural language inference. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 256–262, Minneapolis, Minnesota. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Alexis Conneau, German 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 the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. 1996. Using the framework. Technical report.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. The fourth pascal recognizing textual entailment challenge. *Journal of Natural Language Engineering*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 1180–1189. JMLR. org.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Aria Haghighi, Andrew Ng, and Christopher Manning. 2005. Robust textual inference via graph matching. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Sanda Harabagiu and Andrew Hickl. 2006. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 905–912. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Pratik Joshi, Somak Aditya, Aalok Sathe, and Monojit Choudhury. 2020. TaxiNLI: Taking a ride up the NLU hill. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 41–55, Online. Association for Computational Linguistics.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Nelson F Liu, Roy Schwartz, and Noah A Smith. 2019. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2171–2179.

Bill MacCartney. 2009. *Natural language inference*. Stanford University.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

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

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 216–223, Reykjavik, Iceland. European Languages Resources Association (ELRA).

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Abdelrhman Saleh, Tovly Deutsch, Stephen Casper, Yonatan Belinkov, and Stuart Shieber. 2020. Probing neural dialog models for conversational understanding. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 132–143, Online. Association for Computational Linguistics.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534.

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*.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE.

Eva Vanmassenhove, Jinhua Du, and Andy Way. 2017. Investigating 'aspect'in nmt and smt: Translating the english simple past and present perfect. *Computational Linguistics in the Netherlands Journal*, 7:109–128.

Sara Veldhoen, Dieuwke Hupkes, and Willem H Zuidema. 2016. Diagnostic classifiers revealing how neural networks process hierarchical structure. In *CoCo@ NIPS*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International*

Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4387–4397.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2870–2880, Hong Kong, China. Association for Computational Linguistics.

John Wieting and Douwe Kiela. 2018. No training required: Exploring random encoders for sentence classification. In *International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

F Zanzotto, Alessandro Moschitti, Marco Pennacchiotti, and M Pazienza. 2006. Learning textual entailment from examples. In *Second PASCAL recognizing textual entailment challenge*, page 50. PASCAL.

Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium. Association for Computational Linguistics.

Dataset	# Train	# Dev	# Test
NOISE	20000	5000	5000
UNCORRELATED	20000	5000	5000
PARTIAL	20000	5000	5000
FULL	20000	5000	5000
ATTACKER	23732	5000	5000

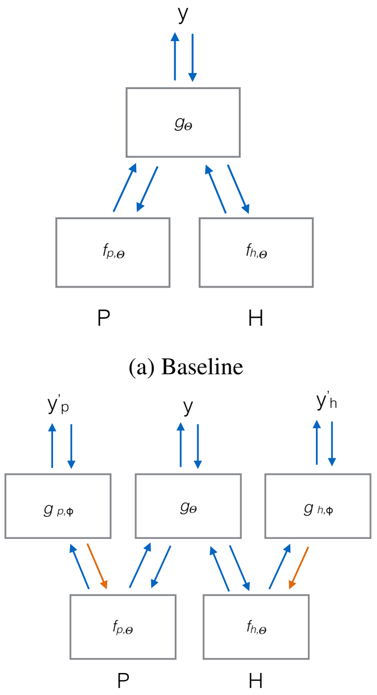
Table 7: Number of train/dev/test examples in constructed synthetic datasets.

A Adversarial Learning Framework

We explore an adversarial framework, as a potential approach to identifying incidentally-encoded properties. We study the utility of this framework within the controlled setting of the synthetic task described in Section 5, where a hypothesis H is entailed by a premise P, if they both begin with the same letter 'a' or 'b'.

We train an adversarial classifier to suppress task-irrelevant information, in this case the presence of 'c'. The goal is to analyze whether adversarial learning can help a model ignore this information while maintaining task performance. If the model succeeds, it indicates the model does not need the linguistic property for the task. Table **??** provides descriptive statistics for Noise, Uncorrelated, Partial and Full synthetic datasets, as well as the probing dataset used to train the external attack classifier. We ensure that datasets do not have any data leakage in the form of strings appearing across train/dev/test splits, or across task and probing splits in either the main task or the external held-out attacker dataset.

We follow the adversarial learning framework illustrated in Figure 3. In this setup, we have premise-hypothesis pairs $\langle p_1, h_1\rangle ... \langle p_n, h_n\rangle$ and entailment labels $y_1 ... y_n$, as well as labels for linguistic properties in each premise–hypothesis pair $\langle z_{p,1}, z_{h,1}\rangle ... \langle z_{p,n}, z_{h,n}\rangle$. We would like to train sentence encoders f(p_i, θ) and f(h_i, θ) and a classification layer g_θ such that $y_i = g_\theta$ (f(p_i, θ), f(h_i, θ)), in a way that does not use $\langle z_{p,i}, z_{h,i}\rangle$. We do this by incorporating an *adversarial* classification layer g_ϕ such that $\langle z_{p,i}, z_{h,i}\rangle$ = $\langle g_\phi$(f(p_i, θ)), g_ϕ(f(h_i, θ)$\rangle$ (Goodfellow et al., 2014; Ganin and Lempitsky, 2015). Following Elazar and Goldberg (2018), we also have an external 'attacker' classifier ϕ' to predict $z_{p,i}$ and $z_{h,i}$ from the learned sentence representation.[10] A similar setup has been used by Belinkov et al. (2019b) to remove hypothesis-only biases from NLI models.



(a) Baseline

(b) Adversarial removal.

Figure 3: Illustration of (a) The baseline NLI task architecture, and (b) Adversarial removal of linguistic properties from the representations. Arrows represent direction of propagation of inputs in the forward pass and gradients in backpropagation. Blue and orange arrows correspond to the gradient being preserved and reversed respectively.

In training the adversarial classifier is trained to predict z from the sentence representations $f_\theta(p_i, h_i)$, and the sentence encoder f is trained to make the adversarial classifier unsuccessful at doing so. This is operationalized through the following training objectives optimized jointly:

$$\arg\min_{\phi} L(g_\phi(f(p_i, \theta), z_{p,i})) + L(g_\phi(f(h_i, \theta), z_{h,i})) \qquad (1)$$

$$\arg\min_{f,\theta} L(g_\theta(f_\theta(p_i, h_i)), y_i) - (L (g_\phi(f(p_i, \theta), z_{p,i}) + L(g_\phi(f(h_i, \theta), z_{h,i}))) \qquad (2)$$

where L is cross-entropy loss. The optimization is implemented through a Gradient Reversal Layer (Ganin and Lempitsky, 2015) g_λ which is placed between the sentence encoder and the adversarial classifier. It acts as an identity function in the forward pass, but during backpropogation scales the

[10] We train the attacker on a held-out dataset with the linguistic property distributed as random noise (Table **??**). We also ensure all examples in the attacker data are unseen in the main task, to prevent data leakage.

	Noise			Uncorrelated			Partial			Full		
	Dev	Adv.	Attack.	Dev	Adv.	Attack.	Dev	Adv.	Attack.	Dev	Adv.	Attack.
Majority	50.4	51.2	50.2	50.94	74.31	50.2	50.62	99.82	50.2	55.34	55.34	50.2
λ=0.0	100.0	-	90.3	100.0	-	93.6	100.0	-	91.08	100.0	-	100.0
λ=0.5	100.0	47.81	95.3	100.0	70.36	62.26	100.0	99.31	80.48	100.0	51.23	93.42
λ=1.0	100.0	49.43	94.5	100.0	71.28	74.1	100.0	99.79	68.8	100.0	52.37	92.58
λ=1.5	100.0	42.7	100.0	100.0	71.54	99.1	97.98	99.79	82.32	100.0	49.8	97.58
λ=2.0	100.0	46.19	99.36	100.0	70.62	99.98	100.0	94.83	91.12	100.0	40.94	94.64
λ=3.0	100.0	46.98	94.64	100.0	70.92	99.8	99.26	99.19	79.66	100.0	53.08	87.0
λ=5.0	99.98	38.87	96.92	99.94	71.0	86.6	100.0	98.73	100.0	100.0	51.32	98.74

Table 8: Adversarial performance on synthetic tasks: noise, uncorrelated, partial, full. Dev is accuracy of model on task, Adv. is accuracy of the adversarial classifier, Atttack. is accuracy of attacker classifier on held-out data.

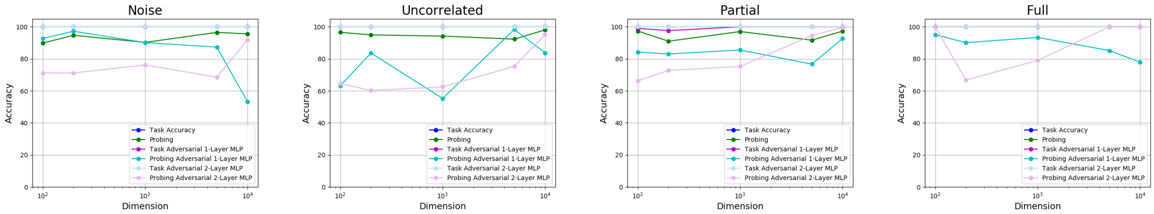


Figure 4: Main Task and Attacker Accuracy as a function of capacity of adversarial classifier for $\lambda = 0.0$ and $\lambda = 1.0$.

gradients by a factor $-\lambda$ [11], resulting in the objective:

$$\arg\min_{f,\theta} L(g_\theta(f_\theta(p_i,h_i)),y_i) + L(g_\phi(g_\lambda(f(p_i \\ ,\theta))),z_{p,i}) + L(g_\phi(g_\lambda(f(h_i,\theta))),z_{h,i}) \quad (3)$$

Implementation details : We implemented the adversarial model using the Dynet framework (Neubig et al., 2017), with a BiLSTM architecture of hidden dimension 200 units. Fixed length vector representations are constructed using the last hidden state and the model is trained for 10 epochs using early stopping. The attacker classifier is a 1-layer MLP with hidden size of 200 dimensions.

Results Table 8 reports the performance of the adversarial and attacker classifiers on the four test sets. We observe that information about irrelevant properties can be extracted by attackers even under adversarial suppression, consistent with the findings of Elazar and Goldberg (2018) and Belinkov et al. (2019b). In the case of random noise, we do not find any setting of adversary weight λ that suppresses the attribute. We further explore issues of adversarial classifier strength:

Adversarial classifier capacity: A more powerful adversarial classifier may be more effective at suppressing task-irrelevant information. To examine this, we fix the attacker classifier and experiment with adversarial classifiers with 1-layer and 2-layer MLP probes and dimensions 100, 200, 1000, 5000 and 10000 units, as reported in Figure 4. We find that varying the capacity of the adversarial classifier can decrease the attacker accuracy, though the choice of capacity depends on the setup used.

Considerations: 1) In synthetic tests, the main task function is learnable by a neural network. However, in practice for most NLP datasets this might not be true, making it difficult for models to reach comparable task performance while suppressing correlated linguistic properties. 2) Information might be encoded, but may still not be recoverable by the choice of probing classifier[12]. 3) Adversarial learning does not remove all information from the representation (Elazar and Goldberg, 2018). 4) If comparable task accuracy can't be reached, one cannot conclude a property is not relevant.[13]

[11] λ controls the extent to which we try to suppress the property.

[12] All claims related to probing task accuracy, as in most prior work, are with respect to the probing classifier used.

[13] This could be because the main task might be more complex to learn or unlearnable, or multiple alternate confounds could be present in data which are not representative of the decision-making needed for the main task.