Do Language Embeddings Capture Scales?

Xikun Zhang[*†]
Stanford University
xikunz2@cs.stanford.edu

Deepak Ramachandran[*]
Google Research
ramachandrand@google.com

Ian Tenney
Google Research
iftenney@google.com

Yanai Elazar
Bar Ilan Univesity, AI2
yanaiela@gmail.com

Dan Roth
University of Pennsylvania
danroth@seas.upenn.edu

Abstract

Pretrained Language Models (LMs) have been shown to possess significant linguistic, common sense and factual knowledge. One form of knowledge that has not been studied yet in this context is information about the scalar magnitudes of objects. We show that pretrained language models capture a significant amount of this information but are short of the capability required for general common-sense reasoning. We identify contextual information in pre-training and numeracy as two key factors affecting their performance, and show that a simple method of canonicalizing numbers can have a significant effect on the results. [1]

1 Introduction

The success of contextualized pretrained Language Models like BERT (Devlin et al., 2018) and ELMo (Peters et al., 2018) on tasks like Question Answering and Natural Language Inference, has led to speculation that they are good at Common Sense Reasoning (CSR). (Especially on twitter)

On one hand, recent work has approached this question by measuring the ability of LMs to answer questions about physical common sense (Bisk et al., 2020) ("How to separate egg whites from yolks?"), temporal reasoning (Zhou et al., 2020) ("How long does a basketball game take?"), and numerical common sense (Lin et al., 2020). On the other hand, after realizing some high-level reasoning skills like this may be difficult to learn from a language-modeling objective only, (Geva et al., 2020) injects numerical reasoning skills into LMs by additional pretraining on automatically generated data. All of these skills are prerequisites for CSR.

Commonly used methods to evaluate CSR

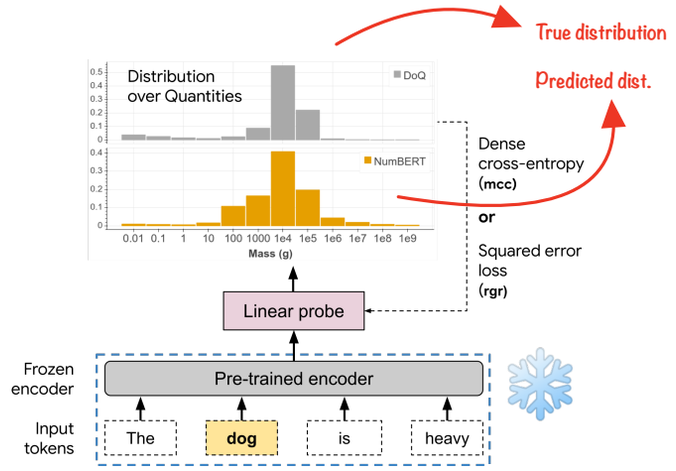

Figure 1: Scalar probing example. The mass of "dog" is a distribution (gray histogram) concentrated around 10-100kg. We train a linear model over a frozen (shown by the snowflake in the figure) encoder to predict this distribution (orange histogram) using either a dense cross-entropy or a regression loss (Section 3).

What does "mass" represent here????

Here, we address a simpler task which is another pre-requisite for CSR: the prediction of scalar attributes, a task we call *Scalar Probing*. Given an object (such as a "wedding ring") and an attribute with continuous numeric values (such as Mass or Price), can an LM's representation of the object predict the value of that attribute? Since in general, there may not be a single correct value for such attributes due to polysemy ("crane" as a bird, versus construction equipment) or natural variation (e.g. different breeds of dogs), we interpret this as a task of predicting a distribution of possible values for this attribute, and compare it to a ground truth distribution of such values. An overview of this scalar probing is shown in Figure 1. Examples of ground-truth distributions and model predictions for different objects and attributes are shown in Figure 2.

Our analysis shows that contextual encoders, like BERT and ELMo, perform better than noncontextual ones, like Word2Vec, on *scalar probing* de-

[*] Both authors contributed equally.
[†] Work done during an internship at Google Research.
[1] Code and models are available at https://github.com/google-research-datasets/numbert.

- Aims to tackle a task: predicting scalar attributes. This is a simple task and a pre-requisite for Common Sense Reasoning (CSR)
- The algo looks simple (yet to get into the details): Freeze the encoder, add a linear layer, and predict the attribute for a given object.
- The attribute can be anything depending on the object, so the attribute values are predicted as the values of a possible distribution

- Intuition says that anything that can perform well on contextual tasks will perform well on non-contextual as well? But is that really true?
- If the contextual models like BERT outperform noncontextual models like Word2vec, is it due to the fact that the BERT has "context" aware training or is it rather just that BERT is more expressive than Word2Vec? Let's find out...

Very very interesting!

spite the task being non-contextual (Mikolov et al., 2013). Further, we show that using scientific notation to represent numbers in pre-training can have a significant effect on results (though sensitive to the evaluation metric used). Put together, these results imply that scale representation in contextual encoders is mediated by transfer of magnitude information from numbers to nouns in pre-training and making this mechanism more robust could improve performance on this and other CSR tasks. We also show improvements by zero-shot transfer from our probes to 2 related tasks: relative comparisons (Forbes and Choi, 2017) and product price prediction (Jianmo Ni, 2019), indicating that our results are robust across datasets.

2 Problem Definition and Data

We define the scalar probing task (see Figure 1) as the problem of predicting a distribution over values of a scalar attribute of an object. We map these values into 12 logarithmically-spaced buckets, so that our task is equivalent to predicting (the distribution of) the order of magnitude of the target value. We explore both models that predict the full distribution and models that predict a point estimate of the value, which is essentially a distribution with all the mass concentrating on one bucket.

Our primary resource for the scalar probing task is Distributions over Quantities (DoQ; Elazar et al., 2019) which consists of empirical counts of scalar attribute values associated with >350K nouns, adjectives, and verbs over 10 different attributes, collected from web data. In this work, we focus only on nouns (which we refer to as *objects*) over the scalar attributes (or *scales*) of MASS (in grams), LENGTH (in meters) and PRICE (in US Dollars). For each object and scale, DoQ provides an empirical distribution over possible values (e.g. Figure 2) that we map into the 12 afore-mentioned buckets and treat it as "ground truth". We note that DoQ itself is derived heuristically from web text and itself contains noise; however, we use it as a starting point to evaluate the performance of different models. Moreover, we validate our findings with transfer experiments shown in Section 6, using DoQ to train a probe that is evaluated on the ground-truth data of Forbes and Choi (2017) and Jianmo Ni (2019).

To explore the role of context in scalar probing, we also trained specialized probing models on a subset of DoQ data in narrow domains: MASS of Animals and PRICE of Household products.

3 Probing Model

We probe three different embedding models: Word2vec (Mikolov et al., 2013), ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) (the latter two of which are contextualized encoders). For each encoder, the input layer extracts an embedding of the object and the probing layer predicts the scalar magnitude. [2]

Input representations For Word2vec, we follow the standard practice of averaging the embeddings of each word in the object's name. If an object name is a full phrase in the dictionary, we use its embedding instead. As BERT and ELMo are contextual text encoders operating on full sentences, we generate artificial sentences with the following templates:

- **MASS:** `The X is heavy.`
- **PRICE:** `The X is expensive.`
- **LENGTH:** `The X is big.`

Templates

and use the CLS token emebedding (for BERT) or final state embedding (for ELMo) as the input representation. For LENGTH, We use "big" instead of "long", since LENGTH measurements in DoQ can be widths or heights as well. Variations of these templates with different adjectives and sentence structures (e.g. "The X is small." or "What is the length of X?" for LENGTH) led to very similar performance in our evaluations.

Probes We use linear probes in all cases following many previous probing work (Shi et al., 2016; Ettinger et al., 2016; Pimentel et al., 2020) since we want to use a simple probe to find easily accessible information in a representation. Hewitt and Liang (2019) also demonstrates that linear probes achieve relatively high selectivity compared to non-linear ones like MLP.

We experiment with two different approaches for predicting scales:

Regression (rgr) For the point estimate, we use a standard Linear Regression model trained

[2] We use Word2Vec embeddings of dimension size 500 trained on Wikipedia, BERT-Base (L=12, H=768, A=12, Total Parameters=110M) trained on Wikipedia+Books and ELMo-Small (LSTM Hidden Size=1024, Output Size=128, #Highway Layers=1, Total Parameters=13.6M) trained on the 1 Billion Word Benchmark, approximately 800M tokens of news crawl data from WMT 2011.

- For point estimate task, regression makes sense but why MCC for the full distribution? Well, predicting a bucket is much easier than predicting the actual value over the full distribution. You can try it yourself. Try rating things in the range 1-100, first by predicting a specific value , and then by predicting the buckets (10-20, 20-30, ...)
- For a task related to CSR, it doesn't matter if you predict a scalar value compared to a bucket. For example, it doesn't matter if you rate a perform 61 or 68. What matters more is that prediction range/bucket (60-70) is as accurate as possible

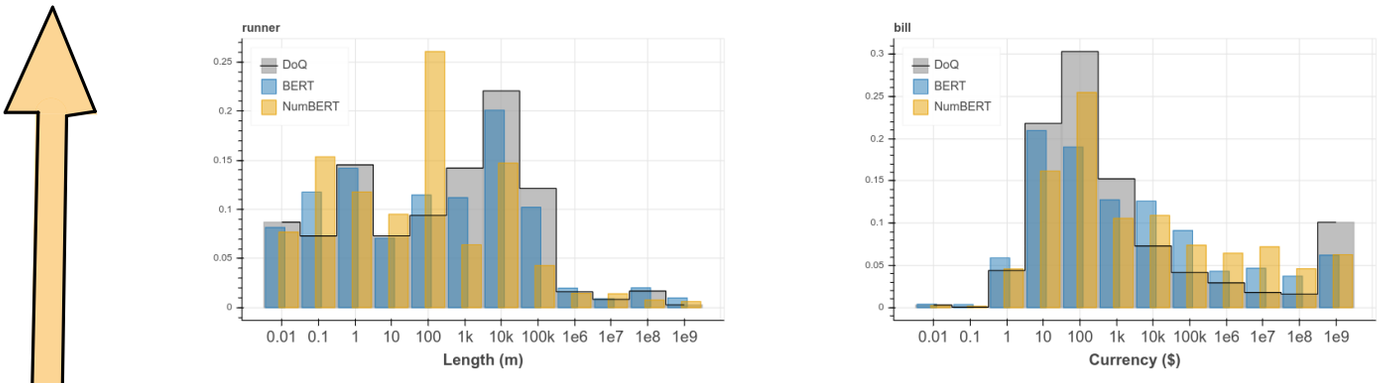

Figure 2: Empirical DoQ distributions and scalar probe predictions for MCC+BERT and MCC+NumBERT (Section 4). The left plot shows length for the term 'runner', showing two peaks corresponding to the length of runner cloths and distances run in races. The right plot shows price for the term 'bill', with counts corresponding to popular denominations and the volumes of larger currency transactions.

to predict log of the median of all values for each object for the scale attribute under consideration.

Multi-class Classification (mcc) We take a non-parametric approach to modeling the full distribution of scalar values and treat the prediction of which bucket a measurement will fall under as a multi-class classification task, with one class per bucket. A similar approach was shown by (Van Oord et al., 2016) to perform well for modeling image pixel values. This approach discards the relationship between adjacent bucket values, but it allows us to use the full empirical distribution as soft labels. We train a linear model with softmax output, using a dense cross-entropy loss against the empirical distribution from DoQ.

More details of the model and training procedure are in the Appendix.

4 Numeracy through Scientific Notation

Wallace et al. (2019) showed that BERT and ELMo had a limited amount of *numeracy* or numerical reasoning ability, when restricted to numbers of small magnitude. Intuitively, it seems that significant model capacity is expended in parsing the natural representation of numbers as Arabic numerals, where higher and lower order digits are given equal prominence. As further evidence of this, it is shown in Appendix B of Wallace et al. (2019) that the simple intervention of *left-padding* numbers in ELMo instead of the default *right-padding* used in Char-CNNs greatly improves accuracy on these tasks.

To examine the effect of numerical representations on scalar probing, we trained a new version of the BERT model (which we call NumBERT) by replacing every instance of a number in the training data with its representation in *scientific notation*, a combination of an *exponent* and *mantissa* (for example `314.1` is represented as `3141[EXP]2` where `[EXP]` is a new token introduced into the vocabulary). This enables the BERT model to more easily associate objects in the sentence directly with the magnitude expressed in the exponent, ignoring the relatively insignificant mantissa. This model converged to a similar loss on the original BERT Masked LM+NSP pre-training task and a standard suite of NLP tasks (See Appendix) as BERT-base, demonstrating that it was not over-specialized for numerical reasoning tasks.

5 Evaluation

We offer the following **aggregate** baseline to help interpret our results: For each attribute, we compute the empirical distribution over buckets across all objects in the training set, and use that as a predicted distribution for all objects in the test set (this is a stronger version of the majority baseline used in classification tasks). Since we are comparing results from regression and classification models, we report results on 3 disparate metrics that give a full picture of performance:

Predicted value is mapped to a single bucket and then accuracy is evaluated in this case

		Accuracy		MSE		EMD	
		mcc	rgr	mcc	rgr	mcc	rgr
Lengths	**Aggregate**	.24	.24	.027	.027	.077	.077
	word2vec	.30	.12	.026	.099	.079	.072
	ELMo	**.43**	.23	**.019**	.084	.055	.072
	BERT	.42	.24	.020	.084	.056	.072
	NumBERT	.40	.22	.021	.086	**.052**	.072
Masses	**Aggregate**	.15	.15	.026	.026	.076	.076
	word2vec	.26	.20	.025	.088	.082	.077
	ELMo	**.36**	.21	**.021**	.087	.061	.077
	BERT	.33	.22	**.021**	.085	.062	.077
	NumBERT	.32	.20	**.021**	.088	**.057**	.077
Prices	**Aggregate**	.24	.24	.019	.019	.057	.057
	word2vec	.26	.14	.019	.090	.063	.087
	ELMo	**.37**	.21	**.016**	.081	.051	.087
	BERT	.33	.19	.017	.083	.054	.087
	NumBERT	.32	.17	.017	.085	**.051**	.087
Animal Masses	**Aggregate**	.30	.30	.022	.022	.059	.059
	word2vec	.33	.35	.021	.069	.069	.077
	ELMo	**.43**	.28	**.016**	.079	.057	.077
	BERT	.41	.26	.017	.079	.058	.077
	NumBERT	.42	.23	.018	.083	**.053**	.077

Table 1: Comparison of encoders and probes on the Scalar probing task on DoQ data. Results are averaged over 10-fold cross-validation.

① **Accuracy** For **mcc** we use the highest scoring bucket from the predicted distribution as the predicted bucket, while for **rgr** we map the predicted scalar to the single containing bucket and use that as the predicted bucket. Then the accuracy is calculated between the predicted bucket and the ground-truth bucket, which is the highest scoring bucket in the empirical distribution in DoQ.

② **Mean Square Error (MSE)** When used to compare distributions, this is also known as the *Cramer-Von Mises* distance (Baringhaus and Henze, 2017). It ignores the difference in magnitude between different buckets (a difference in probability mass between buckets i and $i+1$ is equivalent to the same difference between buckets i and any other), but is upper-bounded by 1, making it easier to interpret. To calculate MSE for **rgr**, we assume that it assigns a probability of 1 to the single containing bucket.[3]

Different from normal reg model evaluation

③ **Earth Mover's Distance (EMD)** Also known as the *Wasserstein* distance (Rubner et al., 1998).

Given two probability densities p_1 and p_2 on Ω, and some distance measure d on Ω, the Earth Mover's Distance is defined as follows:

$$D(p_1, p_2) = \inf_{\pi} \int_{\Omega} \int_{\Omega} \mathrm{d}(x, y) d\pi(x, y)$$

where the infimum is over all non-negative measures π on $\Omega \times \Omega$ satisfying $\pi(E \times \Omega) - \pi(\Omega \times E) = \int_E p_1(x)dx - \int_E p_2(x)dx$ for measurable subsets $E \subset \Omega$. Intuitively, EMD measures how much "work" needs to be done to move the probability mass of p_1 to p_2, while MSE measures pointwise what the difference in densities is. So EMD accounts for the distance between buckets, and predictions to neighboring buckets are penalized less than those further away.

EMD is favored in the statistics literature because of its better convergence properties (Rubner et al., 1998), and there is evidence that it is more robust to adversarial perturbations of the data distribution (Liu et al., 2019), which is relevant for our transfer tasks described below.

Transfer experiments We also evaluate models trained on DoQ on 2 datasets containing ground truth labels of scalar attributes. The first is a human-labeled dataset of *relative comparisons* (e.g. *(person, fox, weight, bigger)*) (Forbes and Choi, 2017).

[3]This is distinguished from the MSE loss used to train regression models, as it is a distance measure over pairs of distributions.

Predictions for this task are made by comparing the point estimates for **rgr** and highest-scoring buckets for **mcc**. The second is an empirical distribution of product price data extracted from the Amazon Review Dataset (Jianmo Ni, 2019). We retrained a model on DoQ prices using 12 power-of-4 buckets to support finer grained predictions.

6 Results

Table 1 shows results of scalar probing on DoQ data.[4] For **MSE** and **EMD** the best possible score is 0, while for accuracy we take a loose upper bound to be the performance of a model that samples from the ground-truth distribution and is evaluated against the mode. This method achieves accuracies of 0.570 for lengths, 0.537 for masses, and 0.476 for prices. Compared to the baseline, we can see that **mcc** over the best encoders capture about half (as measured by accuracy) to a third (by **MSE** and **EMD**) of the distance to the upper bound, suggesting that while a significant amount of scalar information is available, there is a long way to go to support robust commonsense reasoning.

From Table 1, we see that the more expressive models using **mcc** consistently beat **rgr**, with the latter frequently unable to improve upon the Aggregate baseline. This shows that scale information is present in the embeddings, but training on the median alone is not enough to reliably extract it;

[4]The full set of experimental results are shown in Table 6 in the Appendix.

Why median estimation isn't sufficient enough to extract the scaling information from the embeddings?
PS: Can you think of any other estimate that is sufficient enough to extract those details?

the full data distribution is needed.

So our intuition was correct!

Comparing results by encoders, we see that Word2Vec performs significantly worse than the contextual encoders – even though the task is non-contextual – indicating that contextual information during pre-training improves the representation of scales.

Attention is "not all" you need 🤪😝

Despite being weaker than BERT on downstream NLP tasks, ELMo does better on scalar probing, consistent with it being better at numeracy (Wallace et al., 2019) due to its character-level tokenization.

NumBERT does consistently better than ELMo and BERT on the **EMD** metric, but worse on **MSE** and Accuracy. This is in contrast to other standard benchmarks such as Q/A and NLI, where NumBERT made no difference relative to BERT. Our key takeaway is that the numerical representation has an impact on scale prediction (see Figure 2 for qualitative differences), but the direction is sensitive to the choice of evaluation metric. As discussed in Section 5, we believe EMD to be the most robust metric *a priori*, but this finding highlights the need to still examine the full range of metrics.

This is again intuitive.

Results on Animal Masses (Table 1) show that training models only on objects in a narrow domain can significantly improve scalar prediction, underscoring the importance of context. For example, while "crane" in general can refer to either a bird or a piece of construction equipment, only the former is relevant in the animal domain, giving the model a simpler distribution of masses to predict.

Note that, despite significant differences in the raw numbers for each scale (mass/length/price), the relative behavior of encoders, metrics and probes are the same, indicating that our conclusions are broadly applicable.

Transfer experiments On the F&C relative comparison task (Table 2), **rgr**+NumBERT performed best, approaching the performance of using DoQ as an oracle, though short of specialized models for this task (Yang et al., 2018). Scalar probes trained with **mcc** perform poorly, possibly because a finer-grained model of predicted distribution is not useful for the 3-class comparative task. On the Amazon price dataset (Table 3) which is a full distribution prediction task, **mcc**+NumBERT did best on all three metrics. On both zero-shot transfer tasks, NumBERT was the best encoder on all configurations of metric/objective, suggesting that manipulating numeric representations can significantly improve performance on scalar prediction.

	dev		test	
	mcc	rgr	mcc	rgr
word2vec	.40	.73	.38	.74
ELMo	.47	.71	.47	.72
BERT	.48	.71	.49	.71
NumBERT	.51	.77	.54	.76
DoQ [Elazar et. al. 2019]	-	.78	-	.77
Yang et. al. '18	-	**.86**	-	**.87**

Table 2: Accuracy on VerbPhysics (Forbes and Choi, 2017).

	Accuracy		MSE		EMD	
	mcc	rgr	mcc	rgr	mcc	rgr
Aggregate	.04	.04	**.02**	**.02**	.06	.06
word2vec	.09	.23	**.02**	.07	.07	.08
BERT	.14	.25	**.02**	.07	.06	.08
NumBERT	.18	**.27**	**.02**	.07	**.05**	.08

Table 3: Results on consumer price data (Jianmo Ni, 2019).

7 Conclusion

From our novel scalar probing experiments, we find there is a significant amount of scale information in object embeddings, but still a sizable gap to overcome before LMs achieve this prerequisite of CSR. We conclude that although we observe some non-trivial signal to extract scale information from language embedding, the weak signals suggest these models are far from satisfying common sense scale understanding.

Our analysis points to improvements in modeling context and numeracy as directions in which progress can be made, mediated by the transfer of scale information from numbers to nouns. The NumBERT intervention has a measurable impact on scalar probing results, and transfer experiments suggest that it is an improvement. For future work we would like to extend our models to predict scales for sentences bearing relevant context about scalar magnitudes, e.g. "I saw a baby elephant".

Acknowledgments

We want to thank Daniel Spokoyny for the idea of using scientific notation for numbers and Jeremiah Liu for helpful discussions on statistical distance measures.

References

L Baringhaus and N Henze. 2017. Cramér–von mises distance: probabilistic interpretation, confidence intervals, and neighbourhood-of-model validation. *Journal of Nonparametric Statistics*, 29(2):167–188.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

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*.

Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. 2019. How large are lions? inducing distributions over quantitative attributes. In *Association for Computational Linguistics (ACL)*.

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.

Maxwell Forbes and Yejin Choi. 2017. Verb physics: Relative physical knowledge of actions and objects. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 266–276, Vancouver, Canada. Association for Computational Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. *arXiv preprint arXiv:2004.04487*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*.

Julian McAuley Jianmo Ni, Jiacheng Li. 2019. Justifying recommendations using distantly-labeled reviews and fined-grained aspects. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models. *arXiv preprint arXiv:2005.00683*.

Hong Liu, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. 2019. Transferable adversarial training: A general approach to adapting deep classifiers. In *Proceedings of the 36th International Conference on Machine Learning*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 59–66. IEEE.

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.

Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. 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 5306–5314, Hong Kong, China. Association for Computational Linguistics.

Yiben Yang, Larry Birnbaum, Ji-Ping Wang, and Doug Downey. 2018. Extracting commonsense properties from embeddings with limited human guidance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 644–649, Melbourne, Australia. Association for Computational Linguistics.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In *Association for Computational Linguistics*.

A Model Hyperparameters

Here we provide the model hyperparameters we use for reproducibility.

A.1 Probing Layer of the Scalar Probing Model

For the regression model, we use a ridge regression with regularization strength of 1. For the multi-class classification model, we use a linear classifier with a softmax activation function and regularization strength of 0.01.

Task	Metric	BERT Base	NumBERT
CoLA	Dev Acc	.745	.742
MNLI	Dev Matched Acc	.791	.789
	Dev Mismatched Acc	.795	.798
MRPC	Dev Acc	.816	.802
Squad v1	F1	.799	.789
Squad v2	Best F1	.669	.673
STS-B	Dev Pearson's r	.866	.871

Table 4: NumBERT vs BERT-base on a suite of standard NLP benchmarks.

Dataset	Subset	#Data Samples
DoQ	all masses	76,424
	all prices	212,277
	all lengths	244,517
	animal masses	519
	product category prices	1,789
Product Price	-	1,888
F&C Cleaned	train	172
	dev	1,267
	test	1,522

Table 5: Statistics of Datasets/Resources used in our paper

For experiments on the narrow domains with smaller datasets, we first use PCA to reduce embeddings down to 150 dimensions before training the probing model.

A.2 NumBERT

NumBERT is pretrained on the Wikipedia and Books corpora used by the original BERT paper (Devlin et al., 2018). The BERT configuration is the same as BERT-Base (L=12, H=768, A=12, Total Parameters=110M). The language model masking is applied after WordPiece tokenization with a uniform masking rate of 15%. Maximum sequence length (number of tokens) is 128. We train with batch size of 64 sequences for 1,000,000 steps, which is approximately 40 epochs over the 3.3 billion word corpus. All the other hyperparameters and implementation details (optimizer, warm-up steps, etc.) are the same as the original BERT implementation. Table 4 shows a comparison of NumBERT vs a re-implementation of BERT-base with identical settings as above, on a suite of standard NLP benchmarks, and we conclude that the two models reach similar performance on these tasks.

B Data Statistics

Table 5 shows the statistics of 3 datasets/resources we use in this paper. For DoQ, we take the original resource and get each subset by filtering using the corresponding dimensions and/or object types (e.g. all objects, animals, product categories, etc.). Also, only objects with more than 100 values collected in the resource are used. For F&C Cleaned dataset, we use the data and the train/dev/test splits from (Elazar et al., 2019).

C Complete Experimental Results

We model the distributions of those scalar attributes as categorical distributions over 12 categories. We first take the base-10 logarithm of all the values and then round them to the nearest integer (between -2 and 9 for all scales). We treat each integer as a bucket and use the normalized counts in each bucket as the true distribution for that scalar attribute of the object.

To explore the effect of ambiguity, we divide all the data in DoQ for each scale into 2 sets, **Unimodal** where the distribution has one well-defined peak and **Multimodal**, where multiple peaks are present. The number of peaks were identified by a simple hill-climbing algorithm.

As words often have more than one meaning in different contexts or even modifiers, their corresponding distribution from DoQ should reflect the different senses if they appeared enough in the data. When the objects are different enough (e.g. an ice-cream have mainly one meaning and its size doesn't vary much, as opposed to a truck which can be a toy truck, which is very small, or an actual vehicle, which is very big), they may have different modalities. In order to better understand our results, we wish to separate between objects of different modalities to objects with a single modality.

In order to estimate a multi-modal function, we take the bucketed DoQ distribution and smooth it into a probability density function. Then, by finding local maxima over the fitted density function, we estimate a distribution to be multi-modal if we find more than one maximum, otherwise we determine it to be a single-modal distribution.

The complete experiment results including the mutlimodal experiments are in Table 6.

			Accuracy			MSE			EMD		
			All	Multi.	Uni.	All	Multi.	Uni.	All	Multi.	Uni.
Lengths	**mcc**	**Aggregate**	.240	.250	.230	.027	.028	.025	.077	.078	.075
		word2vec	.300	.310	.280	.026	.022	.031	.079	.074	.087
		ELMo	.430	.420	.440	.019	.020	.017	.055	.056	.053
		BERT	.420	.410	.420	.020	.021	.018	.056	.058	.054
		NumBERT	.400	.400	.410	.021	.022	.019	.052	.053	.049
	rgr	**Aggregate**	.240	.250	.230	.027	.028	.025	.077	.078	.075
		word2vec	.120	.120	.130	.099	.100	.097	.072	.070	.074
		ELMo	.230	.230	.240	.084	.085	.082	.072	.070	.074
		BERT	.240	.230	.240	.084	.085	.081	.072	.070	.074
		NumBERT	.220	.210	.220	.086	.088	.084	.072	.070	.074
Masses	**mcc**	**Aggregate**	.150	.150	.150	.026	.027	.024	.076	.077	.074
		word2vec	.260	.260	.260	.025	.026	.023	.082	.083	.080
		ELMo	.360	.360	.360	.021	.021	.019	.061	.062	.059
		BERT	.330	.330	.330	.021	.022	.019	.062	.063	.060
		NumBERT	.320	.320	.330	.021	.022	.019	.057	.058	.055
	rgr	**Aggregate**	.150	.150	.150	.026	.027	.024	.076	.077	.074
		word2vec	.200	.190	.200	.088	.090	.086	.077	.076	.080
		ELMo	.210	.200	.210	.087	.088	.085	.077	.076	.080
		BERT	.220	.210	.220	.085	.086	.084	.077	.076	.080
		NumBERT	.200	.190	.200	.088	.089	.086	.077	.076	.080
Prices	**mcc**	**Aggregate**	.240	.240	.250	.019	.021	.016	.057	.060	.054
		word2vec	.260	.250	.280	.019	.014	.024	.063	.055	.072
		ELMo	.370	.360	.380	.016	.018	.013	.051	.053	.047
		BERT	.330	.320	.330	.017	.019	.014	.054	.055	.051
		NumBERT	.320	.320	.330	.017	.019	.014	.051	.053	.048
	rgr	**Aggregate**	.240	.240	.250	.019	.021	.016	.057	.060	.054
		word2vec	.140	.130	.150	.090	.093	.085	.087	.084	.092
		ELMo	.210	.210	.220	.081	.083	.078	.087	.084	.092
		BERT	.190	.190	.190	.083	.085	.081	.087	.084	.092
		NumBERT	.170	.180	.170	.085	.087	.083	.087	.084	.092
Animals Masses	**mcc**	**Aggregate**	.300	.280	.330	.022	.021	.024	.059	.055	.064
		word2vec	.330	.320	.350	.021	.020	.023	.069	.066	.075
		ELMo	.430	.440	.420	.016	.015	.019	.057	.056	.059
		BERT	.410	.390	.450	.017	.016	.019	.058	.057	.060
		NumBERT	.420	.430	.410	.018	.016	.020	.053	.052	.055
	rgr	**Aggregate**	.300	.280	.330	.022	.021	.024	.059	.055	.064
		word2vec	.350	.350	.360	.069	.069	.069	.077	.081	.070
		ELMo	.280	.250	.330	.079	.080	.077	.077	.081	.070
		BERT	.260	.260	.240	.079	.076	.085	.077	.081	.070
		NumBERT	.230	.230	.240	.083	.081	.086	.077	.081	.070
Household Product Prices	**mcc**	**Aggregate**	.470	-	-	.010	-	-	.046	-	-
		word2vec	.510	.490	.540	.008	.008	.008	.041	.041	.041
		ELMo	.540	.520	.570	.007	.007	.007	.038	.038	.039
		BERT	.570	.560	.580	.007	.007	.007	.038	.038	.039
		NumBERT	.550	.530	.570	.007	.007	.007	.038	.038	.039
	rgr	**Aggregate**	.470	-	-	.010	-	-	.046	-	-
		word2vec	.450	.430	.480	.056	.058	.055	.092	.094	.090
		ELMo	.420	.400	.460	.058	.059	.057	.092	.094	.090
		BERT	.440	.420	.460	.057	.059	.055	.092	.094	.090
		NumBERT	.420	.390	.460	.060	.062	.057	.092	.094	.090

Table 6: Evaluation on all datasets.