EigenNoise: A Contrastive Prior to Warm-Start Representations

Anonymous ACL submission

Abstract

In this work, we present a naïve initialization scheme for word vectors based on a dense, independent co-occurrence model and provide preliminary results that suggests it is competitive, and warrants further investigation. Specifically, we demonstrate through information-theoretic minimum description length (MDL) probing that our model, EigenNoise, can approach the performance of empirically trained GloVe despite the lack of *any* pre-training data (in the case of EigenNoise). We present these preliminary results with interest to set the stage for further investigations into how this competitive initialization works without pre-training data, as well as to invite the exploration of more intelligent initialization schemes informed by the theory of harmonic linguistic structure. Our application of this theory likewise contributes a novel (and effective) interpretation of recent discoveries which have elucidated the underlying distributional information that linguistic representations capture from data and contrast distributions.

1 Introduction

Within the last decade, representation learning in NLP has experienced many major shifts, from context-independent word vectors (Mikolov et al., 2013a,b; Pennington et al., 2014), to context-dependent word representations (Howard and Ruder, 2018; Peters et al., 2018), to pre-trained language models (Devlin et al., 2019; Radford et al., 2018, 2019). These trends have been accompanied by large architectural developments from the dominance of RNNs (Hochreiter and Schmidhuber, 1997), to the appearance of attention (Bahdanau et al., 2015) and the proliferation of the Transformer architecture (Vaswani et al., 2017).

Despite gains on empirical benchmarks, recent works suggest surprising findings: word order may not matter as much in pre-training as previously thought (Sinha et al., 2021), random sentence encodings are surprisingly powerful (Wieting and Kiela, 2018), one can replace self-attention operations in BERT (Devlin et al., 2019) with unparameterized Fourier transformations and still retain 92% of the original accuracy on GLUE (Lee-Thorp et al., 2021), and many modifications to the Transformer architecture do not significantly impact model performance (Narang et al., 2021). While there's no denying increases in empirical performance, these confounding results indicate a lack of understanding of these models and the processing needed to perform NLP tasks.

In this work, we take a step back and consider the (slightly older, yet still popular) paradigm of context independent word vector algorithms like GloVe and word2vec. Specifically, we reflect on the relationships between prediction-based, neural methods and co-occurrence matrix factorization, proposing a naive model of co-occurrence which assumes all words co-occur at least once. Such a naive assumption yields a co-occurrence matrix that can be directly computed and used as a representation for words based on their rank-frequency, and we provide preliminary results that indicate that such an approach is surprisingly competitive to an empirically trained model.

2 Background

2.1 Word Vectors as Matrix Factorization

There is a deep connection between word representation algorithms and factorization of co-occurrence matrices. This is transparent in GloVe (Pennington et al., 2014) by definition, as the log-co-occurrence counts are factored in an online fashion by minimizing Eq. 1, with word vectors u, v, bias parameters a, b, and f, a weighting function:

$$\sum_{i,j} f(X_{ij}) \left(\vec{u}_i \vec{v}_j^T + a_i + b_j - \log X_{ij}\right)^2 \quad (1)$$

Similarly, word2vec's skipgram with negative sampling (SGNS) (Mikolov et al., 2013a) has been shown to implicitly factor a co-occurrence distribution's shifted pointwise mutual information (PMI) matrix (Levy and Goldberg, 2014), namely: $\vec{u}_i\vec{v}_j^T \approx \log \frac{X_{ij}M}{x_i y_j k}$, where k is the number of negative samples, M is the total number of co-occurrences, and x_i and y_i are the marginal number of co-occurrences for the i^{th} row and j^{th} column. Critically, word2vec's negative samples lead its vectors to factor a matrix that provides relative information about *how independently* words co-occur (Levy et al., 2015; Salle and Villavicencio, 2019).

Some suggest that contrast helps improve quality, especially for rare words and syntax (Salle and Villavicencio, 2019; Shazeer et al., 2016). Word2vec supposedly differs from GloVe's strict absorption of positive co-occurrences. However, we now know that GloVe's bias vectors seemingly each model X's marginal distributions independently (Kenyon-Dean et al., 2020). Specifically, GloVe's bias terms appear to optimize as $a_i \approx \log x_i$ and $b_j \approx \log y_j$. So while some researchers have noted that GloVe is *under-defined* by only training on positive observations (Shazeer et al., 2016), we now know that GloVe's bias terms essentially learn the missing contrastive information during optimization (Kenyon-Dean et al., 2020). *This means GloVe is roughly equivalent to SGNS-word2vec*. Specifically, while SGNS-word2vec is granted its contrastive information via marginal sampling, GloVe naïvely utilizes bias parameters, which optimize towards the same marginal contrast distributions, independently. This connection has taken time to emerge from the literature, and from it now we ask a research question that is core to this work: how effective is a representation learned from contrastive information, alone?

2.2 Evaluating Representations via Probes

Significant work has gone into understanding the information captured in language representations (Clark et al., 2019; Conneau et al., 2018; Hewitt and Liang, 2019; Tenney et al., 2019; Vig and Belinkov, 2019; Voita et al., 2019). Early work centered on intrinsic and extrinsic properties (Schnabel et al., 2015), the differences between dense and count-based vectors (Baroni et al., 2014; Levy et al., 2015), and the information contained in a single sentence vector (Conneau et al., 2018). Transitioning towards large pre-trained language models has shifted focus towards characterizing what these models are learning to understand the linguistic phenomena captured within learned representations (Hewitt and Liang, 2019) and self-attention maps (Clark et al., 2019; Tenney et al., 2019; Vig and Belinkov, 2019; Voita et al., 2019).

One method of understanding relies on probing a representation by measuring classifier accuracy enabled with a representation (Hewitt and Liang, 2019; Zhang and Bowman, 2018). However, many approaches fail to sufficiently differentiate the properties of learned representations (Voita and Titov, 2020). This is especially apparent with the high performance of random baselines (Wieting and Kiela, 2018; Zhang and Bowman, 2018) and the ability of probes to accurately encode random labels (Hewitt and Liang, 2019).

3 Effective Word Vectors, Sans Data

3.1 Contrast and Co-occurrence

Since LMs can seemingly learn from shuffled data and retain a surprising amount of predictive power (Sinha et al., 2021), it appears that a great deal of information exists in contrastive information on its own. In the context of co-occurrences X, learning from shuffled data is equivalent to learning from independent (co-occurrence) statistics, e.g., via the cross product of X's marginals. While we seek to determine the extent to which independent statistics are behind the predictive power of deep learning algorithms for benchmark NLP applications, we note that PMI *must* be constant-zero on independently-occurring joint distributions (by PMI's definition). Hence, we cannot simply study independent models of data through the lens of standard GloVe or SGNS-word2vec, leading us to exclude bias terms from GloVe. When paired with a model, $\hat{X}$, of independent co-occurrences, this is roughly equivalent to learning an SGNS-word2vec model via strictly contrastive learning information.

Removing GloVe's bias terms also simplifies its analysis. This is further aided by relieving GloVe of its weighting function:

$$\sum_{i,j} \left(\vec{u}_i\vec{v}_j^T - \log X_{ij}\right)^2 \quad (2)$$

and has the effect of de-biasing optimization by row, i.e., un-balancing the learning rates that GloVe had modulated for lower-frequency words in its formulation. This simple form allows us to straightforwardly approach the word embeddings' common objective's underlying matrix-factorization problem, whose analytic solution requires $\min_{ij}\{\hat{X}_{ij}\} > 0$. In other words, provided all word pairs are modeled to co-occur at least once, the loss can easily be solved in closed forms by well-known matrix factorizations, e.g., by an eigen-decomposition. While the positivity of $\hat{X}$ can be ensured without assuming independence, another immediate benefit of studying contrastive (independent) co-occurrence models is the guarantee that they provide for $\hat{X}$'s positivity. Specifically, since $x_i, y_j > 0$ for all i and j in any co-occurrence data, a reasonable constraint on modeling independent co-occurrences requires positivity across all joint frequencies: $\min_{ij}\{\hat{X}_{ij}\} > 0$. This is evident in marginal-cross-products, for which $\min_{ij}\{\hat{X}_{ij}\} = 1$ due to hapax-legomena ubiquity.

3.2 Harmonic Statistical Structure

To avoid the use of *any* data while representing a target task's vocabulary, $\mathcal{W}$, a model of what pre-training *learns* is needed—here, a distributional model of co-occurrence. For documents, marginal distributions of co-occurrences (unigram distributions) can generally be observed to exhibit harmonic structure, i.e., can generally be modeled via Zipf's law (Zipf, 1935, 1949): $\hat{x}_i = N/r_i$. Without loss of generality, the r_i, or, *ranks*, intuitively indicate the number of *other* words which occur at least as often. In this presentation, we likewise scale by $N = |\mathcal{W}|$ to ensure the vocabulary's smallest unigram 'frequency' is 1. This should raise a question of alignment—how to index the target vocabulary's harmonic structure— which we resolve by counting and ranking the target task's training tokens. Necessarily, this makes our representation reliant on *some* empirical information, namely an ordering of the target task's training data by its vocabulary's ranks (r_i).

Now, assuming harmonic unigram frequencies for our model implies the rows of $\hat{X}_{i,j}$ should marginalize according to $\hat{x}$. To model co-occurrences, we self-sample from $\hat{x}$ for $2m\hat{x}_i$ other words to model the sliding window of $\pm m$ words around each token of the modeled document. Since co-occurrences also exhibit hapax legomena, we set $\min_{ij}\{\hat{X}_{ij}\} = 1$, which forces a closed form:

$$\hat{X}_{ij} = \frac{2mN}{r_i r_j H_N}, \tag{3}$$

where H_N is the N^{th} harmonic number.

3.3 Eigen-Decomposing Distributional Noise

While there are many matrix factorization methods that could be applied, a straightforward approach applies the eigen-decomposition of $\hat{X}$. As it turns out, the symmetry of $\hat{X}$ (and any empirical co-occurrence matrix) ensures the existence of a diagonal matrix, Λ, of unique eigenvalues and an invertible eigen-space matrix, Q, that moreover is orthogonal, i.e., with $Q^{-1} = Q^T$. This leads to an eigen-decomposition of the form: $\hat{X} = Q\Lambda Q^T$. This means that the columns of Q are unit vectors—just like a one hot encoding/standard basis set. Like with other matrix factorizations, a dimensionality reduction to $d < N$ dimensions and approximation of $\hat{X}$ can be derived by the removal of the smallest $N - d$ eigenvalues, Λ_d. We retain half of the approximating structure and call it *EigenNoise*.

4 Experimentation

To evaluate the performance of our proposed initialization scheme, we compare our model against a randomly initialized (parameters simply drawn from a standard normal distribution) baseline as well as empirical GloVe word vectors trained on the Gigaword corpus (Pennington et al., 2014). We evaluate performance on tasks selected from two downstream benchmarks: CoNLL2003 (Tjong Kim Sang and De Meulder, 2003) and TweetEval (Barbieri et al., 2020). From CoNLL-2003, we consider Parts-of-Speech (POS) tagging and Named Entity Recognition (NER) as small-scale, token-based classification tasks to quantify a baseline ability to represent these linguistic constructs in a representation space. TweetEval is a sequence classification benchmark designed to test a model's ability to represent and classify tweets (Barbieri et al., 2020).We select 5 of the 7 sub-tasks to explore regularity in social labels: irony (**I**), hate speech (**H**), offensive language (**O**), emotion (**E**), and stance (**S**).

5 Results & Discussion

5.1 CoNLL

Table 1 (Left) presents the results of probing on CoNLL-2003. Consistently, backpropagating through representations reduces the codelength. This isn't surprising; the embedding layer contains the most parameters. However, what is surprising is that EigenNoise starts at high codelengths (indicating poor regularity with respect to the labels),

Gigaword				
m	PoS		NER	
0	**88.1** ± 0.0	**88.3** ± 0.1	**92.4** ± 0.0	**92.2** ± 0.1
2	**89.1** ± 0.0	**91.5** ± 0.0	**95.7** ± 0.1	**95.8** ± 0.1
5	**87.2** ± 0.4	**91.1** ± 0.1	**95.4** ± 0.1	**95.6** ± 0.1
10	**85.0** ± 0.2	**90.5** ± 0.1	**94.9** ± 0.2	**95.3** ± 0.1

EigenNoise				
m	PoS		NER	
0	74.2 ± 0.0	86.5 ± 1.3	83.8 ± 1.2	90.3 ± 0.1
2	64.3 ± 10.4	89.5 ± 0.4	87.3 ± 0.1	93.5 ± 0.1
5	71.2 ± 0.1	89.6 ± 0.2	86.9 ± 0.1	93.9 ± 0.1
10	69.0 ± 0.1	89.6 ± 0.3	86.6 ± 0.1	93.7 ± 0.4

Random				
m	PoS		NER	
0	77.1 ± 0.7	81.2 ± 1.3	85.2 ± 2.8	86.8 ± 1.9
2	69.8 ± 3.1	76.7 ± 1.1	84.8 ± 0.6	90.2 ± 1.4
5	63.1 ± 2.3	84.7 ± 1.6	83.2 ± 1.0	91.4 ± 0.2
10	60.0 ± 0.4	85.6 ± 0.6	83.9 ± 0.2	91.6 ± 0.3

Task	Gigaword	
I	**60.7** ± 0.6	**61.5** ± 0.8
H	**51.3** ± 0.2	51.2 ± 1.2
O	**76.7** ± 0.6	**80.2** ± 0.7
E	**61.2** ± 0.4	66.9 ± 0.8
S	65.7 ± 5.5	64.5 ± 6.3

Task	EigenNoise	
I	51.8 ± 2.2	58.4 ± 1.5
H	47.5 ± 0.2	**52.5** ± 2.5
O	72.9 ± 0.0	76.4 ± 1.9
E	39.7 ± 0.8	**67.6** ± 1.0
S	**66.8** ± 5.7	64.4 ± 4.0

Task	Random	
I	49.1 ± 2.1	52.6 ± 2.2
H	51.2 ± 1.1	50.5 ± 0.3
O	72.9 ± 0.0	72.3 ± 0.2
E	39.5 ± 0.1	41.3 ± 0.7
S	65.7 ± 4.1	**65.6** ± 4.0

Table 1: (Left) Test set accuracy on CoNLL2003 tasks. Accuracy is averaged across random seeds ± the standard deviation, with left and right accuracy for frozen and un-frozen embeddings respectively. m indicates the window size. (Right) Test set accuracy on TweetEval tasks. Accuracy is averaged across random seeds ± the standard deviation, with left and right accuracy for frozen and un-frozen embeddings respectively.

but, when allowed to update, is able to approach the codelengths of empirical GloVe. This suggests that, while EigenNoise isn't quite ideal immediately, if allowed to adapt to the task at hand, it can do so with relatively little data. When factoring in the naivety of EigenNoise, the fact that it can approach the empirical GloVe model that has a far larger vocabulary (400K words versus 20K ranks) and is trained on infinitely more data, these result are more compelling. Other interesting observations include that the theory-based vectors do worse compared to the standard normal random vectors when both are held static. However, when both are allowed to update their representations, the random vectors barely reduce the codelength whereas the theory-based vectors more than halve theirs. At the very least, this indicates the theory-based rank vectors are an interesting weight initialization point.

5.2 TweetEval

Table 1 (Right) displays the results of probing on TweetEval. Here, we observe that the random vectors are clearly the worst overall, but that all these representations perform similarly for these types of tasks, with the empirical GloVe model performing best out-of-the-box. This seems fairly reasonable, given the way each model was constructed. One may also observe that the empirical and theory vectors result in similar codelengths for the hate speech detection and offensive language identification tasks when the theory vectors are allowed to update. This seems to indicate that the theory-based vectors do not contain the regular signal needed to detect these social phenomena initially but that the empirical GloVe vectors do. Seemingly, empirically-based GloVe vectors contain a higher degree of information about hate speech and offensive language out-of-the-box when compared to an EigenNoise set that's free from such biases, yet the latter can adapt through tuning.

6 Conclusion & Future Work

In this work, we introduce an incredibly naive initialization scheme for independent word vectors such as GloVe and word2vec. We provide preliminary experimentation that demonstrates the efficacy of such a scheme in a low-compute setting through an information-theoretic approach with MDL probing. We believe that these preliminary results are interesting and beg further investigation, especially as an initialization scheme for independent word vectors even if they are to be empirically tuned.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. Tweet-Eval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $ &!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

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.

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. Lstm can solve hard long time lag problems. *Advances in neural information processing systems*, pages 473–479.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Kian Kenyon-Dean, Edward Newell, and Jackie Chi Kit Cheung. 2020. Deconstructing word embedding algorithms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8479–8484, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2021. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27:2177–2185.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3111–3119.

Sharan Narang, Hyung Won Chung, Yi Tay, William Fedus, Thibault Fevry, Michael Matena, Karishma Malkan, Noah Fiedel, Noam Shazeer, Zhenzhong Lan, et al. 2021. Do transformer modifications transfer across implementations and applications? *arXiv preprint arXiv:2102.11972*.

Jeffrey Pennington, Richard Socher, and Christopher D 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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alexandre Salle and Aline Villavicencio. 2019. Why so down? the role of negative (and positive) pointwise mutual information in distributional semantics. *arXiv preprint arXiv:1908.06941*.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 298–307.

Noam Shazeer, Ryan Doherty, Colin Evans, and Chris Waterson. 2016. Swivel: Improving embeddings by noticing what's missing. *arXiv preprint arXiv:1602.02215*.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

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.

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

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.

G. K. Zipf. 1935. *The Psycho-Biology of Language*. Houghton-Mifflin.

G. K. Zipf. 1949. *Human Behaviour and the Principle of Least-Effort*. Addison-Wesley.

A Experimental Details for Probing

The probes used in this work are simple multi-layer perceptrons with a single hidden layer, hidden dimension of 512, and no dropout, defined as: $\hat{y}_i \sim \text{softmax}(W_2\text{ReLU}(W_1 h_i))$. For sequence classification tasks, the entire sequence is embedded and then averaged. For token classification tasks, a window size $m \in \{0, 2, 5, 10\}$ is selected and the $2w + 1$ token window is embedded and flattened. Each experiment is repeated 3 times on random seeds $\in \{0, 1234, 322111\}$ with data block splits chosen to align with previous work: 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.25, 12.5, 25, 50, 100 % of the data.

A.1 Representations

We compare three representations: GloVe trained on the GigaWords corpus (Pennington et al., 2014), our EigenNoise model, and a baseline where parameters are sampled from a standard normal distribution. EigenNoise uses a vocab size of $N = 20,000$, just large enough to fit the training vocab of each data set to demonstrate the efficacy of this approach in a "low-compute" setting. For all representations, a dimensionality of 50 is used and both freezing and un-freezing the embedding layer is explored.

A.2 Optimization

All probes are trained with the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate 0.001. Adhering to previous works (Hewitt and Liang, 2019; Voita and Titov, 2020), we anneal the learning rate by a factor of 0.5 once the epoch does not lead to a new minimum loss on the development set; training stops after 4 such epochs.

GigaWord				
m	**PoS**		**NER**	
0	**85.5** ± 9.5	**85.4** ± 0.3	**47.2** ± 0.1	**47.9** ± 0.4
2	**99.2** ± 0.4	**79.7** ± 0.5	**32.7** ± 0.4	**29.4** ± 0.1
5	**121.4** ± 0.7	**91.6** ± 0.4	**38.5** ± 0.3	**33.5** ± 0.6
10	**142.4** ± 1.2	104.4 ± 0.3	**44.8** ± 0.6	**38.6** ± 0.3
EigenNoise				
m	**PoS**		**NER**	
0	221.9 ± 4.5	110.8 ± 0.1	121.7 ± 1.5	66.4 ± 0.2
2	205.1 ± 1.7	90.9 ± 0.3	89.7 ± 0.6	40.5 ± 1.0
5	218.6 ± 0.5	92.4 ± 0.2	91.3 ± 0.2	41.8 ± 0.6
10	239.0 ± 0.6	**96.9** ± 0.5	97.1 ± 0.2	44.7 ± 1.1
Random				
m	**PoS**		**NER**	
0	157.8 ± 3.8	137.7 ± 2.9	95.1 ± 1.8	83.7 ± 0.7
2	197.7 ± 12.4	129.3 ± 1.4	103.5 ± 3.3	62.4 ± 1.0
5	252.5 ± 4.8	138.2 ± 1.8	116.7 ± 2.2	65.4 ± 2.1
10	281.9 ± 10.9	147.1 ± 1.7	125.3 ± 1.8	69.7 ± 1.6

Task	**Gigaword**	
I	**1.9** ± 0.0	**1.9** ± 0.0
H	**5.3** ± 0.1	**5.0** ± 0.1
O	**6.7** ± 0.0	**6.5** ± 0.1
E	**3.3** ± 0.0	**3.1** ± 0.0
S	**0.5** ± 0.1	**0.5** ± 0.1
Task	**EigenNoise**	
I	**1.9** ± 0.0	**1.9** ± 0.0
H	5.9 ± 0.0	**5.0** ± 0.0
O	7.5 ± 0.0	6.8 ± 0.1
E	4.1 ± 0.0	3.5 ± 0.0
S	**0.5** ± 0.1	**0.5** ± 0.1
Task	**Random**	
I	2.0 ± 0.0	1.9 ± 0.0
H	6.0 ± 0.0	5.7 ± 0.1
O	7.5 ± 0.0	7.5 ± 0.0
E	4.1 ± 0.0	4.1 ± 0.0
S	**0.5** ± 0.1	**0.5** ± 0.1

Table 2: (Left) Codelength performance on CoNLL2003 tasks, measured in kilobytes. Codelengths are averaged across random seeds ± the standard deviation, with left and right codelengths for frozen and un-frozen embeddings respectively. m indicates the window size. (Right) Codelength performance on TweetEval tasks. Codelengths are measured in kilobits. Codelengths are averaged across random seeds ± the standard deviation, with left and right codelengths for frozen and un-frozen embeddings respectively.

A.3 Hardware

Experiments were completed using a single NVIDIA Titan V 12GB on our internal cluster. The combination of representations, heterogeneous dataset, and early stopping criteria result in variable length runs, however, the longest single probe run took no more than 2 hours to complete.

B Dimensionality Reduction

To precisely compute the eigen-decomposition dimensionality reduction, define $I_d \in \mathbb{R}^{N\times d}$ to be the first d columns of the N-dimensional identity matrix (I) and let $\Lambda_d \in \mathbb{R}^{d\times N}$ denote the first d rows of the diagonal eigenvalue matrix. The $\hat{X}$-reconstruction equation is then:

$$\hat{X} \approx QI_d(Q\Lambda_d)^T = U_d V_d^T \tag{4}$$

where $U_d = QI_d$ and $V_d = Q\Lambda_d$ are needed to retain the effect of zeroing out Λ's $N - d$ smallest diagonal elements. This reduces the Q-variation into two low-dimensional (d) representations that approximately reconstruct $\hat{X}$. For our purposes, we retain U_d and refer to the solution as *EigenNoise*.

We note that varying choices could be made to handle Λ—it could be multiplied without loss of generality into the U-side, instead of the V-side. But perhaps more interestingly, Λ's values could be rooted—perhaps over $\mathbb{C}$—for a symmetric set, i.e., with $U_d = V_d$ and $U_d, V_d \in \mathbb{C}^{N\times d}$. We speculate that informative variation over $\mathbb{C}$ may exist, but leave the exploration of this to future work.

C Information-Theoretic Evaluation

Here, we adopt an alternate, information-theoretic probing methodology for evaluation that combines the measure of ease of mapping from representation to label space as well as the complexity of the model needed to do so. This method, called Minimum Description Length (MDL) (Voita and Titov, 2020), is concisely described as measuring the regularity of a representation with respect to a set of labels. Specifically, we adopt the online codelength metric (measured in kilobits), where a smaller codelength is indicative of a more regular representation. We adopt this metric as it is more informative than accuracy and is more stable with respect to random initializations and hyperparameter selection.

C.1 MDL Probing

As discussed in the related works, comparing the performance of pre-trained representations can be more subtle than simply training a classifier (i.e., a *probe*) and comparing the attained performance, sometimes giving un-intuitive results such as random baselines performing comparably well to pre-trained ones. To combat this issue, we adopt the information-theoretic approach of Minimum Description Length (MDL) probing (Voita and Titov, 2020), which serves as a measure of the regularity of a representation with respect to a label set. This allows us to quantify how much difficulty a classifier has in achieving a particular level of performance.

In MDL probing, let

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$$

be a dataset where $x_{1:n} = (x_1, x_2, ..., x_n)$ are representations from a model and $y_{1:n} = (y_1, y_2, ..., y_n)$ are the labels of a desired property. Instead of measuring how well a probe can perform this mapping, MDL tasks a probe with learning to efficiently transmit the data using the representation. Using the online codelength metric, assume that two agents (Alice and Bob) agree upon a form of a model $p_\theta(y|x)$ with learnable weights θ, a random weight initialization scheme, and an optimization procedure.

Break points $1 = t_0 < t_1 < ... < t_S = n$ are selected to form data blocks to be transmitted. Alice begins by transmitting $y_{1:t_1}$ using a uniform code, from which both Alice and Bob train a model $p_{\theta_1}(y|x)$ using the first data block $\{(x_i, y_i)\}_{i=1}^{t_1}$. Alice uses that model to transmit the next data block $y_{t_1+1:t_2}$, which is used to train a better model $p_{\theta_2}(y|x)$ to transmit the next block. This continues until all data has been transmitted, resulting in an online codelength computed via $L^{\text{online}}(y_{1:n}|x_{1:n}) = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}}|x_{t_i+1:t_{i+1}})$. As in (Voita and Titov, 2020), probes that learn mappings via fewer data points will have shorter codelengths.