Language Through a Prism: A Spectral Approach for Multiscale Language Representations

Alex Tamkin[†]
Stanford University

Dan Jurafsky
Stanford University

Noah Goodman
Stanford University

Abstract

Language exhibits structure at different scales, ranging from subwords to words, sentences, paragraphs, and documents. To what extent do deep models capture information at these scales, and can we force them to better capture structure across this hierarchy? We approach this question by focusing on individual neurons, analyzing the behavior of their activations at different timescales. We show that signal processing provides a natural framework for separating structure across scales, enabling us to 1) disentangle scale-specific information in existing embeddings and 2) train models to learn more about particular scales. Concretely, we apply spectral filters to the activations of a neuron across an input, producing *filtered embeddings* that perform well on part of speech tagging (word-level), dialog speech acts classification (utterance-level), or topic classification (document-level), while performing poorly on the other tasks. We also present a *prism layer* for training models, which uses spectral filters to constrain different neurons to model structure at different scales. Our proposed BERT + Prism model can better predict masked tokens using long-range context and produces multiscale representations that perform better at utterance- and document-level tasks. Our methods are general and readily applicable to other domains besides language, such as images, audio, and video.

1 Introduction

Language exhibits structure at multiple levels, ranging from morphology at the subword level [1], word meaning at the lexical level [2], coherence and other discourse properties at the clause or sentence level [3, 4, 5], to topical and narrative structures for entire documents [6, 7]. Prior work in NLP has shown how these kinds of structures can be explicitly modeled by representing individual levels of structure [8, 9, 10, 11, 12, 13], multiple levels of structure [14, 15, 16], building hierarchical models that capture structure at the sentence level [17, 18] or between sentences [19, 20], and probing to discover known linguistic levels of structure [21, 22, 23, 24].

We propose a new method for uncovering and learning this kind of structure in representations at every scale, from word meaning to document topics, without drawing on prior linguistic models of specific structural levels like "sentence" or "clause." To do so, we employ tools from spectral analysis, widely used in signal processing and other fields [25] to separate and control information at different timescales. Intuitively, any sequence of values, such as a neuron's activations across input tokens, can be represented as a weighted sum of cosine waves with different frequencies. The weight for a particular frequency indicates the amount of structure in the sequence at that scale: weight on higher frequencies indicates faster changes in the neuron's activation from token to token, while weight on lower frequencies indicates activations that shift more gradually across an input. By removing certain

[†] atamkin@stanford.edu

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

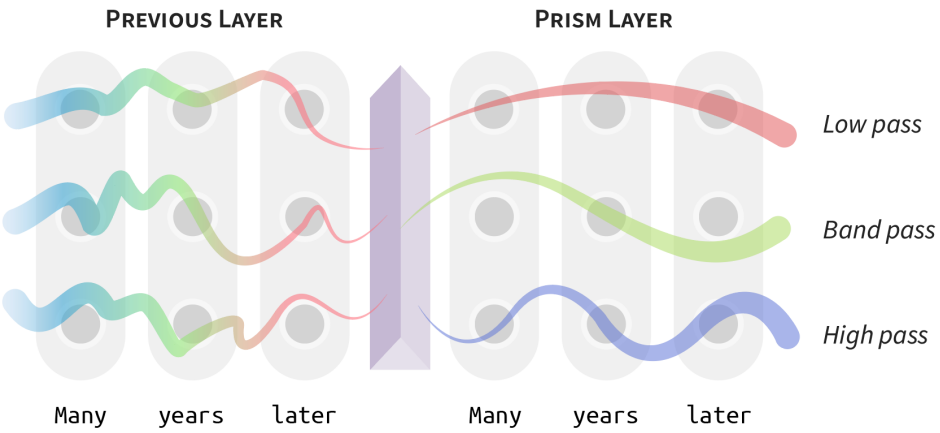

Figure 1: **The prism layer specializes different neurons for different scales.** First, the representations for an input are computed (left; in this case, the input is of length three). Next, a spectral filter (a low-, high-, or band-pass) is applied along the activations of each individual neuron (right). This produces neurons that are only able to represent structure at particular scales. Curved lines illustrate the scales at which neurons can change over an input.

frequencies, called *spectral filtering*, we can remove information about variation at particular scales. See Figure 2 for a visualization.

In this work, we apply spectral filters to the activations of individual neurons in BERT [26], a popular deep NLP model. This enables us to separate information in model representations that changes at different rates across the input—for example, part of speech changes on a word-to-word basis, while topical changes are much more gradual. Concretely, we contribute:

1. **A principled framework** based on spectral analysis for describing structure at multiple scales in deep representations. While we consider applications to NLP models, this is a general framework that could extend to other models with representations arranged in spatial or temporal structure. (Section 2)
2. **A technique**, *spectral filtering*, for extracting scale-specific information from language representations. We show how low-pass filters can alter representations to only perform well on topic classification (document-level), while band-pass and high-pass filters do the same for dialog acts classification (utterance-level) and part of speech tagging (word-level). (Section 3)
3. **A new model component**, the *prism layer*, which specializes neurons in a model for particular scales of structure. After training with a prism layer, our model is more sensitive to long-range interactions between tokens and produces individual representations that perform comparably or better than BERT's across tasks at different scales. (Section 4)

2 Spectral filtering of contextual word representations

This section provides some background on the spectral analysis tools we use and describes how we apply them to deep language representations.

2.1 Background: The discrete cosine transform and spectral filters

In order to perform operations in the frequency domain of a sequence, we first need to obtain a representation of the input in the frequency domain. This is the role of a *spectral transform*. The spectral transform we use in this work is the discrete cosine transform (DCT[2]) [27], a widespread tool used in audio coding, texture analysis, image classification, and compression [28, 27]. The DCT represents a real-valued sequence of points as a same-length sequence of weights over cosine

[2]More precisely, this transform is known as the DCT-II

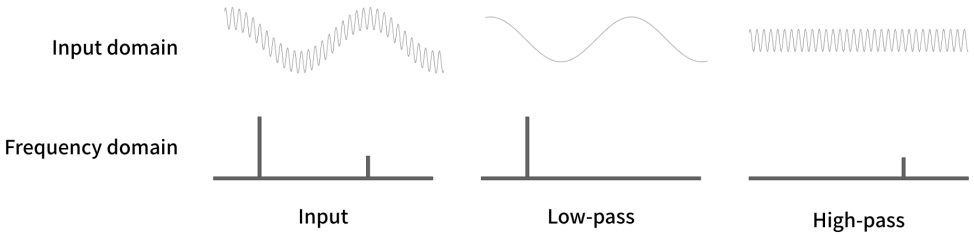

Figure 2: **A visual depiction of spectral filters and their effects in the input and frequency domain.** The input domain shows a sequence of values (e.g., the activation of a neuron across input tokens). The frequency domain shows the weight on the cosine waves which sum to produce the curve in the input domain. Low-pass filters only allow low frequencies to pass through, producing a smoothed input. High-pass filters only allow high frequencies and produce a locally-normalized input. Band-pass filters (not shown) are compositions of low- and high-pass filters.

functions of different frequencies. Formally, for a real-valued sequence $\{x^{(0)} \dots x^{(N-1)}\}$ its DCT (the weights for each frequency) is obtained by

$$f^{(k)} = \sum_{n=0}^{N-1} x^{(n)} \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \qquad k = 0, \dots, N-1 \tag{1}$$

Intuitively, the DCT computes the similarity of a signal and cosine waves of different frequencies by taking the dot product between them. These dot products constitute the coefficients of the signal in the frequency domain. The DCT is closely related to the discrete Fourier transform (DFT). We use the DCT here because it is a real-to-real function (the DFT is complex-to-complex), is widely used in practice, and can often produce fewer artifacts than the DFT when filtering [27, 29].

The DCT of a sequence enables straightforward manipulation of structure at different scales in a sequence. For example, one can remove components above some threshold frequency k_{thresh} by setting $f_k \leftarrow 0$ for all $k > k_{\text{thresh}}$, then applying the inverse DCT (IDCT) to return the signal to the original domain [30]. This is known as a *low-pass filter*, and returns a smoothed, same-length version of the original input, removing shorter-term fluctuations. The inverse operation can be performed to achieve a *high-pass filter*, which returns a signal where each term is locally normalized with respect to its neighbors, neutralizing longer-term trends. Composing these two operations yields a *band-pass filter*, as only a band of frequencies is allowed to pass through the filter. See Figure 2 for a visual depiction.[3]

2.2 Applying the DCT to contextual word representations

How do we apply the DCT to deep language representations? A common feature of modern NLP models is *contextual word representations*, a sequence of vectors created by processing a sequence of tokens (e.g., words or subword units). These representations are produced by a wide range of modern NLP architectures, including Transformer-based [34] models like BERT [26] and GPT-2 [35], as well as LSTM-based [36] models such as ELMo [37].

Assume we are given a sequence of contextual word representations $v_0, \dots, v_{N-1}$. The core technique we propose is to apply the DCT to a slice of these representations *along a single neuron*: $v_0[i], \dots, v_{N-1}[i]$. We refer to the transformed sequence in the frequency domain $f_0[i], \dots, f_{N-1}[i]$ as the spectrum of the ith neuron. $f_0[i]$ is the lowest frequency term, corresponding to the average value of $v_0[i], \dots, v_{N-1}[i]$, while $f_{N-1}[i]$ is the highest frequency term. We can then implement any of the filters from Section 2.1 by zeroing out the appropriate values in the spectrum, and then applying the IDCT to return the sequence to the original domain. In practice, external libraries make this quite simple: we show a three-line implementation of a low-pass filter in Figure 3b.

[3] Fully zeroing out frequencies (a *brick wall filter*) can produce artifacts after performing the IDCT, motivating the use of smoother attenuation functions [31, 32], which reduce artifacts in exchange for allowing less-than-full attenuation of frequencies outside the desired band. However, for simplicity, we use brick wall filters in this work, leaving study of other filters, as well as other spectral tools like wavelets [33], for future work.

Filter	Ex. Scale	Period (toks)	DCT index
HIGH	Word	1–2	130–511
MID-HIGH	Clause	2–8	34–129
MID	Sentence	8–32	9–33
MID-LOW	Paragraph	32–256	2–8
LOW	Document	256–∞	0–1

(a) **The spectral filters we consider in this work**, along with their periods, spectral bands (the indices in the DCT), and example linguistic phenomena at that scale. The period of a cosine wave for a DCT index is the approximate number of tokens it takes for the wave to complete a cycle.

```
def low_pass(H, k):
  H_dct = dct(H.T)
  H_dct[:, k:] = 0
  return idct(H_dct).T
```

(b) **Spectral filters are simple to incorporate into existing models.** Python-style code for a low-pass filter over representations. Input `H` is a list of representations for each input token, while `k` is the low-pass threshold frequency. `T` is the transpose operator. We use a PyTorch library to compute the (I)DCT.

Figure 3

3 The relationship between spectral frequencies and linguistic phenomena

We have seen how to apply spectral filters to the hidden states of deep NLP models. In this section, we explore how these spectral filters can be used to separate out phenomena at different scales in contextual word representations.

3.1 Disentangling scale-specific information in representations

Contextual word representations have been shown to not only encode the meaning of tokens in context [37], but also a wider range of linguistic phenomena such as semantic roles, entity types, constituent labels, relations between entities, and coreference [38]. This suggests that these representations may already be encoding information about multiple scales ranging from the (sub)word itself to its containing phrase, clause, sentence, paragraph and perhaps the document as a whole. In this work, we consider whether these phenomena can be separated out at the level of *individual neurons* by using spectral filters to tease apart structure at different scales in a neuron's activations across an input.

To investigate, we observe how the choice of spectral filter affects the ability of a classifier to perform tasks at different scales using the filtered representations. Each spectral filter is determined by a corresponding *spectral band*: the range of frequencies that is used for the low-, high-, or band-pass. We seek to choose bands corresponding to different scales. However, the scale of a particular frequency is revealed by its period: the number of tokens it takes to complete a full cycle. For example, from Equation 1 we see that index 8 of the DCT has a frequency of $2 * 8 = 16$, and thus for inputs of size 512 has a period of $512/16 = 32$ tokens.

In this work, we divide the frequency spectrum into five bands, chosen reflect the inductive bias that linguistic units at one scale are composed of multiple units from the scale below (e.g. several words compose a phrase). Thus, we allocate bands such that for each band, the periods of the frequencies in the next higher band decay by a fixed amount. This produces five bands (LOW, MID-LOW, MID, MID-HIGH, and HIGH) with a diverse range of scales, as shown in Table 3a.[4] See the Appendix for more details on band allocation and discretization.

3.2 Probing bandpassed representations for linguistic information

We evaluate the content of these filtered representations through probing experiments [39, 40, 41]. For each dataset below, we encode each training example with a fixed, pretrained BERT-Base cased model [26]. This produces a series of 768-dimensional contextual word representations. We then apply a spectral filter along each dimension and train a softmax classifier to perform a particular task using each filtered representation. We examine three English-language tasks, involving classification of word-, utterance-, and document-level phenomena, providing a natural testbed for investigating the content of these representations:

[4] We use these five separate bands in part for instructive purposes; however, in practice, one might wish to smoothly change the endpoints of the spectral band across neurons. One could also specifically choose bands for a task based on their corresponding periods to include or exclude particular scales of interest.

1. **Part of speech tagging (word-level):** We use the Penn Treebank dataset [42]. The task is to predict the part of speech (e.g. PAST TENSE VERB, WH-PRONOUN, CARDINAL NUMBER) from the given token representation.
2. **Dialog speech act classification (utterance-level):** We use the Switchboard Dialog Speech Acts corpus [43, 44, 45].[5] The task is to predict the dialog speech act (e.g. APOLOGY, HEDGE, APPRECIATION) of the utterance containing the given token representation.
3. **Topic classification (document-level):** We use the 20 Newsgroups dataset [46]. The task is to predict the topic (newsgroup; e.g. SCI.SPACE, COMP.GRAPHICS, REC.AUTOS) of the document containing the given token representation.

We train our probing models for a maximum of 30 epochs, using the Adam optimizer [47] with default parameters. We use early stopping with a patience of one, decaying the learning rate by a factor of 2 when successive epochs do not produce a decrease in validation loss. To compare against the masked language modeling (MLM) task, which was the original target task[6] for these representations [26], we also train an MLM probe for three epochs on the WikiText-103 dataset [48].

As Figure 4 shows, different spectral filters indeed produce representations specialized for the expected task. The highest probing accuracy for part of speech tagging occurs when extracting the HIGH band, aligning with the fact that this is a word-level task. However, the highest frequency spectral band still performs worse than the original representations, suggesting that lower frequency information is sometimes necessary for this task (e.g. for parts of speech correlated over several tokens, such as strings of numbers or lists of nouns). By contrast, topic-classification performs best with information from the LOW band, aligning with the fact that it is a document-level phenomenon. Interestingly, the accuracy for the LOW band is substantially higher than for the original representations, suggesting that higher frequency variation present in the original representations may be harmful for that task. Meanwhile, probing for dialog speech acts, a classification task over utterances, is most successful at the MID band, with performance comparable to that of the original representations. The probing results for masked language modeling are most similar to part of speech tagging, underscoring the degree to which MLM is a local task.

These results demonstrate that spectral filters are effective tools for separating multiscale linguistic phenomena in contextual word representations.

4 Using spectral filters during training

In the previous section, we saw how spectral filters can be used to isolate information about linguistic phenomena at different scales in an existing model's representations. However, this observed structure arose naturally from BERT's masked language modeling task, which we saw is a relatively local task. In this section, we will show how spectral filters can be used *during training* to produce multiscale representations with improved performance on mid-scale and global tasks despite being trained with masked language modeling.

4.1 The prism layer

In BERT, the information for the different tasks discussed above may be distributed across all neurons, rather than specialized in particular ones. Spectral filters, however, provide a natural way to force BERT to use different neurons for information about different scales. The resulting multiscale representations may then be better suited for a broader range of tasks than the original BERT representations.

To accomplish this, we take a given hidden state in BERT and divide the units evenly into five *sectors*.[7] To each sector, we then apply a different band-pass from Table 3a. We call these additional computations a *prism layer*, as they separate out the different frequencies in a layer's representations. See Figure 1 for an illustration. In our main experiments, we apply one prism layer after the last BERT layer. See the Appendix for an investigation of placing prism layers after each BERT layer.

[5] We use the preprocessing library from `https://github.com/cgpotts/swda`

[6] We do not consider the next sentence prediction task (NSP) [26]. While it was also used for BERT pretraining, we discard the `[CLS]` tokens, which are used to predict the NSP label.

[7] We distribute the $768 \pmod 5 = 3$ remaining units to the LOW, LOW-MID, and MID bands.

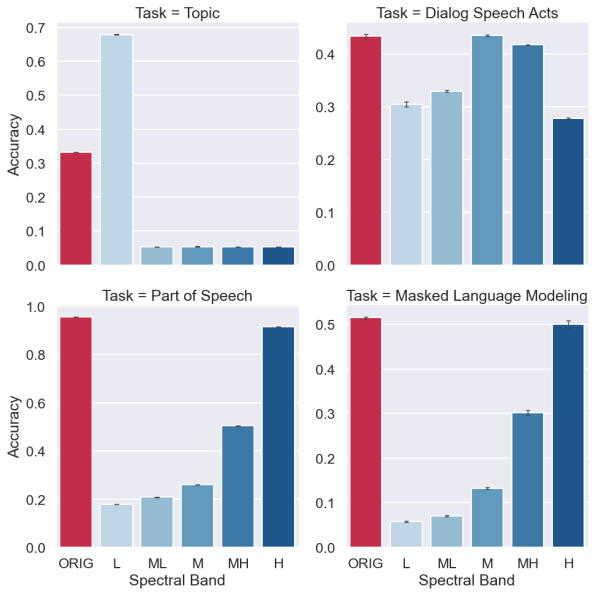

Figure 4: **Different spectral filters extract information useful for tasks at different scales.** Probing accuracy for different tasks and band-passes. A low-pass filter produces representations that yield highest probing accuracy on topic classification, while high-passed representations have highest probing accuracy for part of speech tagging. Meanwhile, band-passing the middle frequencies is most useful for dialog speech act probing. "ORIG" refers to the performance of the original token representations. Error bars show standard deviations over three probing runs.

We then train our pretrained BERT model with the prism layer on the masked language modeling task; this is so the model can adjust to the new constraints imposed upon it and learn to allocate information at particular frequencies to the correct sectors. We use an external PyTorch library for computing and backpropagating through the DCT and IDCT. [8] We train on the WikiText-103 dataset [48] for 50k steps at a batch size of 8 with default parameters for Adam. To allow for fair comparisons between our model and BERT, we also further train an unmodified pretrained BERT model using this same data and procedure (see the Appendix for an ablation of this step).

4.2 Results

We now compare the probing performance of the vanilla BERT model with the BERT model trained with our prism layer, shown in Table 1. The BERT model with the prism layer performs considerably better than BERT on topic (+18.8%) and dialog speech act (+6.9%) classification while maintaining high accuracy on part of speech tagging (-1.5%). These results demonstrate that the prism layer has enabled BERT to produce more general-purpose representations that capture phenomena across scales.

4.3 Sensitivity to distant tokens

The multiscale representations produced by the prism layer are used by the model to perform the masked language modeling (MLM) objective. Since these representations contain information at different scales, this provides an inductive bias for the model to rely on both long-range and short-range information when performing the MLM task. To show this quantitatively, we consider an MLM problem where one hundred consecutive tokens in the middle of the input are masked. The model's loss on these tokens reflects the model's ability to rely on distant information to predict tokens without local context.

We plot the average log probability of the correct token in Figure 5, for both the BERT model with the prism layer, as well as a BERT model trained on WikiText-103 for the same number of steps. As

[8] `https://github.com/zh217/torch-dct`

Table 1: **Training with a prism layer produces multiscale representations that perform comparably or better than BERT across different tasks.** Probing accuracy and standard deviation (3 trials) for different tasks on the final-layer BERT and BERT + Prism representations.

Task	Model	Accuracy (%)	S.D. (%)
Topic classification	BERT	32.21	0.08
	BERT + Prism	**51.01**	0.14
Dialog speech acts	BERT	47.09	0.33
	BERT + Prism	**54.02**	0.61
Part of speech	**BERT**	**95.86**	0.02
	BERT + Prism	94.41	0.02

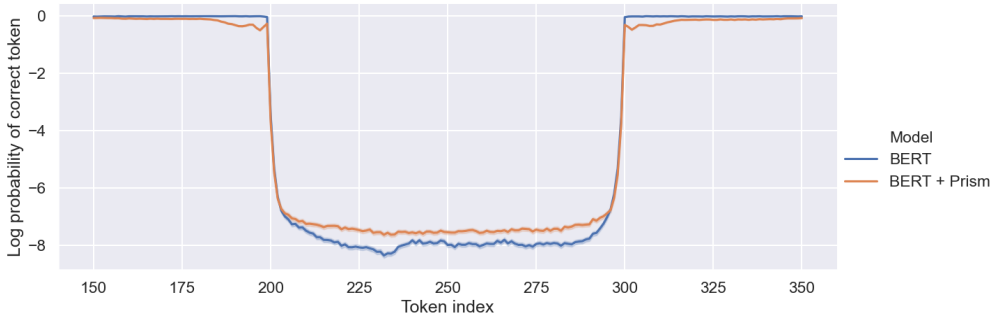

Figure 5: **Training with a prism layer significantly improves prediction of masked tokens without local context (note the log scale).** Average log probability of correct token for different indices (N=1600). Indices between 200 and 300 are replaced with a `[MASK]` token in the input, requiring the model to use long-range context to generate a probability distribution for the missing token. The higher log probabilities in the masked region for the BERT + Prism model suggest the prism layer makes the model more sensitive to long-range dependencies. Shaded regions are 95% bootstrap CIs (generally too small to see without magnification).

expected, no model can precisely guess the missing tokens with perfect accuracy. But we do see a noticeable difference between the probabilities assigned to the correct token by the BERT models with and without the prism layer (note the log scale). This indicates that the BERT + Prism model is a better *long range* language model, using context to predict distant tokens.

Another interesting phenomenon in the graph is the dips in log probability exhibited by the BERT + Prism model adjacent to the redacted text, indicating dependence on (redacted) distant context. No such dip exists for the original BERT model, indicating that it solves the MLM task in a very local way. These results suggest that the prism layer is a useful tool for encouraging modeling of long-range dependencies in Transformer models [49, 36].

5 Related work

Our work connects with several streams of research investigating multiscale structure in natural language and our models of it. Prior work has studied the extent of this structure at different scales in linguistic corpora, using tools ranging from random walk models and power spectra [50, 51] to entropy and mutual information [52]. To model this structure, researchers have conducted multiresolution analyses of text corpora by applying diffusion wavelets to term-document corpora [53], multinomial topic distributions [54], and term-term cooccurrence graphs [55]. Concerning deep learning, several works have considered the challenges of modeling different scales in distributed representations of words [8, 56] and of capturing long-term dependencies in recurrent neural networks [36, 57]. Other work conducts analytic studies of models that illuminate their scale-awareness, including the sensitivity of LSTM language models to relationships at different scales [58] and the attention patterns

of Transformer models [59]. In conversation with this literature, our work provides a principled way of understanding multiscale structure in the representations of deep models, illuminating the linguistic phenomena captured at each of these scales and enabling the construction of scale-specific representations for downstream purposes.

In concert with these analyses, a large body of work has attempted to leverage the expressive capability of distributed representations to improve modeling at particular scales. For example, several works introduce different kinds of architectural modifications to recurrent neural networks in order to encourage learning hierarchical structure, especially long-term structure, including via updating hidden states at different intervals [60], multilayered models [61, 62], incorporating tree structures [17] or syntactic parsing [18], introducing residual connections [63], adding auxiliary losses [49], and discretizing ordinary differential equations [64]. In addition, certain works explicitly focus on creating high-quality representations at particular scales, including the word-level [8, 9], sentence-level [10, 11, 65, 20], paragraph-level [12] and document-level [13]. Perhaps most similar to our work is a stream of work incorporating the Fourier basis into recurrent architectures [66, 67]. However, while these works focus on speeding up training or improving gradient flow in RNNs, our approach is architecture-agnostic, provided the model produces contextual word representations, and can be used to understand or improve specific scales of interest in the model's representations. Another piece of related work is Ordered Neurons [68], which enforces an update hierarchy in the latent state of an RNN to capture tree-like structure in an input (e.g., syntax trees). By comparison, our approach generalizes beyond RNN or autoregressive architectures and can capture both syntactic structure like part of speech as well as longer-range multiscale phenomena like dialog speech acts and topic where tree structures may not be as appropriate.

Finally, our work is related to spectral approaches in audio [69, 70], where it is naturally suited as an input representation, as well as computer vision, where the Fast Fourier Transform [71] and the Discrete Cosine Transform [30] have been used to speed up the training of convolutional neural networks [72], generate filters for scene classification [73], and compress convolutional models [74]. Concerning scales, the authors of StyleGAN [75] investigate how different layers in their model are responsible for phenomena at different scales, such as pose, lighting, face shape, and finer facial features. Most related to our work is a line of research that improves training by using spectral filters to replace downsampling operations in convolutional models [76] as well as improving optimization speed and generalization by removing low-magnitude [77] or high-frequency [76] spectral coefficients. We also explore attenuation of different frequency coefficients, but in an NLP context to improve modeling of long-range dependencies, and further use spectral techniques to understand, control, and improve modeling at different scales.

6 Conclusion

In this work, we demonstrate how techniques from spectral analysis provide a principled and effective framework for separating multiscale phenomena in deep language representations. We first demonstrate how spectral filters can be used to separate information at different scales in BERT representations. We use this technique to produce scale-disentangled representations that perform well at either part of speech tagging, dialog acts classification, or topic classification, while performing poorly on the other two tasks. We also show how to create multiscale representations by training with a prism layer, which forces different neurons to capture information about different scales. The representations produced by the resulting model enable comparable or higher performance across the three tasks than vanilla BERT representations. We also show that training with a prism layer increases the model's sensitivity to long-range context, as measured by a masked language modeling task. These results demonstrate that spectral techniques are a powerful set of tools for uncovering and modeling multiscale phenomena in deep NLP models.

Our work provides multiple avenues for further study. For interpretability researchers, these tools could enable better understanding of knowledge and information processing at different scales in neural models across different tasks, inputs, and layers. For researchers of linguistic change, this method may enable better tracking of topics over time or facilitate the removal of extraneous information (e.g., topic) when targeting a linguistic phenomenon at a different scale. Finally, we also see promise for improving NLP models during training in a broader range of applications and architectures. More generally, we emphasize that our method is domain agnostic: it needs only a collection of representations with some kind of geometric (e.g., spatial or temporal) structure—thus,

we are optimistic about the potential for further applications of these techniques on the hidden states of computer vision, time series, and reinforcement learning models, among others.

Broader Impact

The spectral tools we provide in this paper are applicable to a wide range of neural network models and possible end uses. While this makes it difficult to speak with confidence about broader impacts of the research, we briefly discuss a few potential use cases. Scale isolation enables users to remove information about particular kinds of structure inside existing representations. This could be useful for interpretability or fairness research, as well as computational social scientists who wish to remove e.g. topical information from word embeddings. However, scale isolation may also enable tailored search for particular kinds of information in text or other content, which could enable uses that are beneficial or harmful depending on the use case and whether consent is obtained by relevant parties. The prism layer falls under a general trend of producing more capable neural networks. Such a trend may contribute to increased automation or other changes in labor markets, which may create benefits and harms that depend on the economic and social policies of relevant governing bodies.

Acknowledgments and Disclosure of Funding

We would like to thank Shyamal Buch, Jesse Mu, Shikhar Murty, Ben Newman, Mike Wu, Pratyusha Ria Kalluri, and Jesse Michel for useful discussions and comments on drafts. This work was supported in part by DARPA under agreement FA8650-19-C-7923.

References

[1] Eugene A Nida. Morphology: The descriptive analysis of words. 1949.

[2] D Alan Cruse, David Alan Cruse, D A Cruse, and D A Cruse. *Lexical semantics*. Cambridge university press, 1986.

[3] Andrew Kehler and Andrew Kehler. *Coherence, reference, and the theory of grammar*. CSLI publications Stanford, CA, 2002.

[4] Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225, 1995.

[5] Sandra A Thompson and William C Mann. Rhetorical structure theory: A framework for the analysis of texts. *IPRA Papers in Pragmatics*, 1(1):79–105, 1987.

[6] Marti A Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64, 1997.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

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

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

[10] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[11] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California, June 2016. Association for Computational Linguistics.

[12] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[13] Roberta A Sinoara, Jose Camacho-Collados, Rafael G Rossi, Roberto Navigli, and Solange O Rezende. Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, 163:955–971, 2019.

[14] Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 899–907, 2015.

[15] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.

[16] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[17] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[18] Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*, 2016.

[19] Yangfeng Ji and Jacob Eisenstein. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, 2014.

[20] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[21] Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*, 2018.

[22] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[23] Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*, 2019.

[24] John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. 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 4129–4138, 2019.

[25] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

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

[27] K Ramamohan Rao and Ping Yip. *Discrete cosine transform: algorithms, advantages, applications*. Academic press, 2014.

[28] Chao Zuo, Qian Chen, and Anand Asundi. Boundary-artifact-free phase retrieval with the transport of intensity equation: fast solution with use of discrete cosine transform. *Optics express*, 22(8):9220–9244, 2014.

[29] Alan C Bovik. *The essential guide to video processing*. Academic Press, 2009.

[30] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[31] Stephen Butterworth et al. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.

[32] H Takahasi. On the ladder-type filter network with tchebysheff response. *J. Inst. Elec. Commun. Engrs. Japan*, 34(2):65–74, 1951.

[33] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

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

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

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

[38] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019.

[39] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[40] Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 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, 2016.

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

[42] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. 1993.

[43] Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical Report 97-02, University of Colorado, Boulder Institute of Cognitive Science, Boulder, CO, 1997.

[44] Elizabeth Shriberg, Rebecca Bates, Paul Taylor, Andreas Stolcke, Daniel Jurafsky, Klaus Ries, Noah Coccaro, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3–4):439–487, 1998.

[45] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–371, 2000.

[46] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995.

[47] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[48] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[49] Trieu H Trinh, Andrew M Dai, Minh-Thang Luong, and Quoc V Le. Learning longer-term dependencies in rnns with auxiliary losses. *arXiv preprint arXiv:1803.00144*, 2018.

[50] Werner Ebeling and Alexander Neiman. Long-range correlations between letters and sentences in texts. *Physica A: Statistical Mechanics and its Applications*, 215(3):233–241, 1995.

[51] Alexey N Pavlov, Werner Ebeling, Lutz Molgedey, Amir R Ziganshin, and Vadim S Anishchenko. Scaling features of texts, images and time series. *Physica A: Statistical Mechanics and its Applications*, 300(1-2):310–324, 2001.

[52] Werner Ebeling and Thorsten Pöschel. Entropy and long-range correlations in literary english. *EPL (Europhysics Letters)*, 26(4):241, 1994.

[53] Ronald R Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006.

[54] Chang Wang and Sridhar Mahadevan. Multiscale analysis of document corpora based on diffusion models. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[55] Vidit Jain and Jay Mahadeokar. Short-text representation using diffusion wavelets. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 301–302, 2014.

[56] Aakash Sarkar and Marc Howard. Scale-dependent relationships in natural language. *arXiv preprint arXiv:1912.07506*, 2019.

[57] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[58] Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*, 2018.

[59] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

[60] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. *arXiv preprint arXiv:1402.3511*, 2014.

[61] Salah El Hihi and Yoshua Bengio. Hierarchical recurrent neural networks for long-term dependencies. In *Advances in neural information processing systems*, pages 493–499, 1996.

[62] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

[63] Boxuan Yue, Junwei Fu, and Jun Liang. Residual recurrent neural networks for learning sequential representations. *Information*, 9(3):56, 2018.

[64] Bo Chang, Minmin Chen, Eldad Haber, and Ed H Chi. Antisymmetricrnn: A dynamical system view on recurrent neural networks. *arXiv preprint arXiv:1902.09689*, 2019.

[65] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2016.

[66] Y Zhang and Lai-Wan Chan. Forenet: fourier recurrent networks for time series prediction. 2000.

[67] Jiong Zhang, Yibo Lin, Zhao Song, and Inderjit S Dhillon. Learning long term dependencies via fourier recurrent units. *arXiv preprint arXiv:1803.06585*, 2018.

[68] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. *arXiv preprint arXiv:1810.09536*, 2018.

[69] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP)*, pages 4277–4280. IEEE, 2012.

[70] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[71] Henri J Nussbaumer. The fast fourier transform. In *Fast Fourier Transform and Convolution Algorithms*, pages 80–111. Springer, 1981.

[72] Michael Mathieu, Mikael Henaff, and Yann LeCun. Fast training of convolutional networks through ffts. *arXiv preprint arXiv:1312.5851*, 2013.

[73] Salman H Khan, Munawar Hayat, and Fatih Porikli. Scene categorization with spectral features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5638–5648, 2017.

[74] Yu Cheng, Felix X Yu, Rogerio S Feris, Sanjiv Kumar, Alok Choudhary, and Shi-Fu Chang. An exploration of parameter redundancy in deep networks with circulant projections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2857–2865, 2015.

[75] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[76] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2449–2457. Curran Associates, Inc., 2015.

[77] Salman H Khan, Munawar Hayat, and Fatih Porikli. Regularization of deep neural networks with spectral dropout. *Neural Networks*, 110:82–90, 2019.

[78] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[79] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[80] Wen-Hsiung Chen, CH Smith, and SC Fralick. A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on communications*, 25(9):1004–1009, 1977.

A Spectral Band Allocation

We have n frequencies we wish to allocate into k bands such that $\sum_i a_i = n$, where a_i denotes the number of frequencies allocated to band i.

We begin by allocating each band one frequency, leaving $(n-k)$ frequencies remaining. Next, we choose a base b (we use $b = 4$) and generate allocation scores $s_i = b^i$ which we normalize into fractional allocations $\tilde{a}_i = (n-k)b^i / \sum_i (s_i)$. However, allocations must be whole numbers, so we produce conservative allocations $\check{a}_i = \lfloor \tilde{a}_i \rfloor$, and then allocate the remaining frequencies in descending order of $\tilde{a}_i - \check{a}_i$ (i.e. whichever bands were closest to receiving another frequency) until $\sum_i a_i = n$.

However, we note again that in practice, one might wish to smoothly vary the endpoints of the spectral band between neurons, rather than choosing only 5 canonical bands. In addition, one could specifically choose bands for a task based on their corresponding periods to include or exclude particular scales of interest.

B Additional Ablations

We present additional ablations here, with new results displayed in **bold**. All results are averages of three trials.

B.1 Individual sectors of the BERT + prism model

What information is accessible from individual sectors of the BERT + prism model? For the lowest frequency sector of the BERT + prism model, topic classification probing accuracy is **45.1%**, versus **5.3%** for the highest-frequency sector and 51.0% for the full model. For the highest-frequency sector, POS tagging probing accuracy is **84.1%**, versus **16.8%** for the lowest-frequency sector and 94.4% for the full model. This suggests that the HIGH and LOW frequency bands are largely but not entirely responsible for the BERT + prism model's performance on POS tagging and topic classification, respectively.

B.2 Performance of BERT representations without finetuning on WikiText-103

To confirm that the additional pretraining on WikiText-103 does not harm BERT, producing an artificially weak baseline, we compare probing performance on the original pretrained BERT model. The original model achieves an accuracy of **94.6%** for POS tagging, **41.8%** for dialog acts, and **28.9%** for topic classification, slightly worse than the better model that was trained longer on WikiText-103 (95.9%, 47.1%, 32.2% respectively).

B.3 Placing prism layers after every BERT layer

Can we obtain better representations by adding a prism layer after *each* BERT layer instead of just the last? We find that this model produces worse representations than BERT + prism, achieving **45.2%** accuracy on topic classification (-5.8%), **51.8%** on dialog speech acts (-2.2%) and **94.0%** on POS tagging (-0.4%). We suspect this may be because the removal of spectral information deeper in the network reduces the model's effective capacity, interfering with its ability to produce useful representations.

C Data and preprocessing

We tokenize all data inputs using a WordPiece tokenizer [78] from an external library [79]. For each gold label (e.g., a part of speech tag, a dialog speech act annotation, or a topic), we then perform our probing experiments on each representation from the resulting N tokens, weighting the loss and accuracy for each embedding's prediction by $1/|N|$.

For dialog speech acts and masked language modeling, which have long inputs, we chunk each input into segments of at most length 510, then append the two special tokens before feeding them into our models. For part of speech tagging and topic classification, we discard excess tokens. We use the traditional train/validation/test splits for all models:

- The 20 Newsgroups dataset has approximately 20k documents, with 60% for training and the remainder for testing. `http://qwone.com/~jason/20Newsgroups/`
- The Switchboard Dialog Speech Acts corpus contains around 1.1k training transcripts and 19 validation transcripts. `https://github.com/cgpotts/swda`
- The Penn Treebank has approximately 38.2k examples for training and 5.5k for validation. `https://catalog.ldc.upenn.edu/LDC99T42`
- Wikitext 103 has around 500MB of text for training and 1.1MB for validation. `https://blog.einstein.ai/the-wikitext-long-term-dependency-language-modeling-dataset/`

D Complexity

The DCT can be computed efficiently using the FFT in $O(N \log N)$ [80].

E Computational Time and Resources

All experiments were performed on single Titan XP GPUs. Each experiment took approximately one day, while MLM training with the prism layer took approximately 8 hours. Collecting losses for Figure 5 took on the order of minutes.