Visually Analyzing Contextualized Embeddings

Matthew Berger*

Vanderbilt University

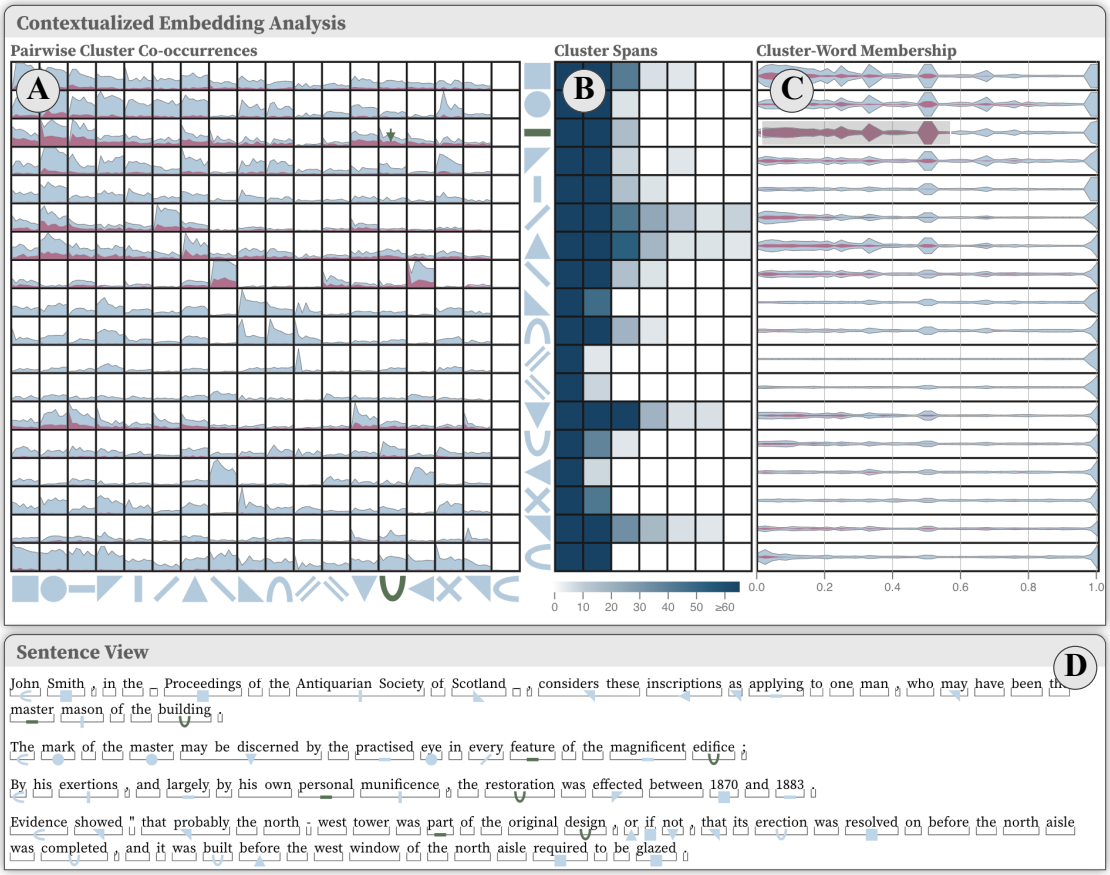

Figure 1: Our method allows the exploration of contextualized embeddings produced by language models. Our design shows (A) co-occurrences of phrases via their assigned clusters, (B) per-cluster span lengths and (C) how much context a given cluster captures. One may also inspect example sentences in detail (D), here highlighting terms that describe building structures.

Abstract

In this paper we introduce a method for visually analyzing contextualized embeddings produced by deep neural network-based language models. Our approach is inspired by linguistic probes for natural language processing, where tasks are designed to probe language models for linguistic structure, such as parts-of-speech and named entities. These approaches are largely confirmatory, however, only enabling a user to test for information known a priori. In this work, we eschew supervised probing tasks, and advocate for unsupervised probes, coupled with visual exploration techniques, to assess what is learned by language models. Specifically, we cluster contextualized embeddings produced from a large text corpus, and introduce a visualization design based on this clustering and textual structure – cluster co-occurrences, cluster spans, and cluster-word membership – to help elicit the functionality of, and relationship between, individual clusters. User feedback highlights the benefits of our design in discovering different types of linguistic structures.

Index Terms: Machine Learning—visual analytics—Natural Language Processing;

1 Introduction

Recent advances in natural language processing (NLP) have led to the development of language models that perform remarkably well across a wide range of language understanding tasks [8, 24], e.g. named entity recognition, entailment, paraphrase verification [37]. These models typically take the form of deep neural networks that are *pre-trained* on a large corpus of unannotated text, and subsequently *fine-tuned* for specific language understanding tasks. An intriguing property of these models is that, due to the combination of the pre-training objective and model capacity, they encode a variety of linguistic structure, despite never being explicitly trained to learn such structure [6, 18, 27]. However, comprehending the full space of *what is learned* is elusive, and remains an open problem.

*e-mail: matthew.berger@vanderbilt.edu

Approaches for interpreting pre-trained language models have relied on the design of *supervised probes* – human-annotated datasets that capture known semantic or syntactic properties, e.g. parts-of-speech, chunking, dependency syntax [2, 18]. Representations extracted from language models are trained to solve problems posed by these probes to assess how well the model captures linguistic structure. Although supervised probes have helped shed light on language models, they inherit several limitations. First, they are confirmatory, only telling us whether or not a language model has learned a known linguistic property. Secondly, models trained to solve probes face issues regarding complexity, e.g. an overly-complex model that performs well may poorly reflect the probe task [13].

In this work we propose an interactive approach to understanding deep, pre-trained, language models. Our work is inspired by existing probing methods, but instead approaches language model interpretability in an unsupervised manner: rather than build probe-specific classifiers, we aim to let the data distribution speak for itself. Specifically, we focus on *contextualized embeddings* of words: vector representations that encode the context of a particular word with respect to its originating sentence. Given a large text corpus, in analogy to supervised probes we *cluster* the embeddings. Given the clustering, the key goal of our visualization is to help a user understand the functionality of clusters, and relationships between clusters. As shown in Fig. 1, our visualization is designed to highlight patterns of linguistic properties: (A) co-occurrences in clusters, (B) formation of phrases via contiguous cluster spans, (C) just how contextual is a given word, as well as (D) details-on-demand for showing individual sentences and their words' cluster assignments. Combined, these views are designed to help the user identify specific linguistic properties through a set of supported interactions.

To evaluate our method we gathered feedback from users to assess what information they could gain by using our system. Through the feedback, we find that different types of linguistic structures, e.g. parts-of-speech, noun phrases, named entities, can be identified through our visualization design.

2 Related Work

Our work is most related to interpretability approaches within, both, NLP and visual analytics for understanding language models.

Neural network-based language models date back to Bengio et al. [3], and have gained recent attention with more sophisticated network architectures and language modeling objectives [8, 24]. These models have demonstrated significant performance gains in a wide variety of language understanding tasks [25, 37], despite the seemingly irrelevant tasks used for pre-training, e.g. masked word prediction and next sentence prediction [8]. This has motivated the design of supervised probes [2, 7] as a way to test what linguistic knowledge language models encode in their learned representations [14, 16, 18, 33]. Yet these methods face several limitations. As supervised models are usually trained from these representations to assess the accuracy of a probing task, overparameterized models might poorly reflect the linguistic knowledge encoded by the language model [13]. Further, it is delicate to design a probing dataset that ensures task relevance in what is learned [9, 26]. Our approach is inspired by probing methods, but is focused on unsupervised methods for interpreting pre-trained language models, complemented by interactive visualization techniques.

Significant work has been developed within the visual analytics community for interpreting deep NLP models, please see Hohman et al. [15] for a broader survey on deep learning and visual analytics, and Spinner et al. [30] for model interpretability within visualization. Visualization methods have been developed to understand context-independent word embeddings, through assessing analogies [19], customizing embedding projections [21] and comparing embeddings [5, 12]. Closely related to our method are approaches that visually analyze recurrent neural networks, namely LSTMVis [32] and RNNVis [22]. RNNVis similarly clusters hidden representations of RNNs, but focuses on specific tasks, e.g. sentiment analysis, whereas we consider task-independent pre-training objectives. Other works have considered the interpretation and editing of sequence-to-sequence models [31], models designed for natural language inference [20], and interactively performing abstractive summarization [10]. Further methods have visually analyzed self-attention in language models [23, 35], whereas we consider contextualized embeddings in Transformer models [34]. Recent work such as Checklist [29] and TX-Ray [28] permit the customization of supervised and unsupervised probes, respectively. In contrast to Rethmeier et al., which focuses on interpreting individual neurons, we consider the embedding space as a whole.

3 Objectives and Tasks

Before discussing the tasks that we aim to support, we first discuss the language model, and extracted representations, used in this work. Our goal is to understand the representations learned by different layers in the Transformer model [34], pre-trained on large amounts of raw textual data using the BERT objectives of masked word, and next sentence, prediction [8]. Specifically, we use the cased 12-layer BERT model of Devlin et al. [8], where for a fixed layer, given a sentence composed of m words $(w_1, w_2, \ldots, w_m)$, passing this sequence through the model provides us with a $d = 768$ dimensional vector for each word, denoted $x(w_j) \in R^d$ for the j'th word in the sentence. We denote $x(w_j)$ as the *contextualized embedding* for word w_j. Note that the same word's contextualized embeddings from two different sentences will likely be different, due to sentence context, e.g. "handle" can be treated as a noun or a verb.

We would like to gain insight on the linguistic properties learned by contextualized embeddings. However, to circumvent the issues inherent in supervised probes, and empower the user in exploration, we approach this in an unsupervised manner. Specifically, given a sentence drawn from a large input corpus, we first obtain the contextualized embedding for each word in the sentence. For a word broken into subwords, the last subword's embedding is taken as the original word's embedding [18]. Next, we cluster the contextualized embeddings over all sentences, using k-means. For robustness, we adapt the initialization scheme of Arthur et al. [1] by limiting seed vectors to unique words, and performing k-means over different initializations, taking the result with lowest sum-of-squared distances to assigned cluster centers. Empirically, we find this scheme produces stable clusters, in part due to the large number of vectors provided by each of our tested corpora [36], e.g. ranging from 75K to 250K vectors. We set the number of clusters, k, to 50 in all experiments.

For a given sentence i and a word at position j in the sentence, we obtain a cluster label $l(w_j^i) \in [1, k]$. The resulting clustering can be viewed as a *proxy* for a set of supervised probes, e.g. one cluster could reflect the verb part-of-speech, while another cluster could represent location-based named entities. However, unlike supervised probes, we do not know, a priori, the meaning of the clusters. Hence, the main purpose of our visualization design is to help the user in understanding (1) *what a cluster represents*, and (2) the *relationships between clusters*. The tasks supported in our design aim to address these objectives, and serve to *abstract* typical approaches taken in supervised probes:

(**T1**) **Assess how much context a given cluster contains.** Certain words (e.g. punctuation) are less reliant on context than other words (e.g. "place") that may have multiple senses. This task intends to abstract multiple probes such as parts-of-speech [18], semantic role labeling [4], and word tense [7].

(**T2**) **Determine a cluster's ability to form meaningful phrases.** This task abstracts segmentation probes such as syntactic chunking and named entity extraction [18], as well as constituency parsing [17].

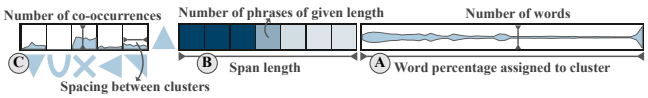

Figure 2: Overview of our design showing (A) relative amount of context encoded by a cluster's set of words (B) frequency over different span lengths for cluster-specific words forming contiguous spans, and (C) cluster co-occurrence frequency regarding word/phrase spacing.

(**T3**) **Analyze relationships between clusters.** This task abstracts relationships between clusters, e.g. relation extraction [18], syntactic dependencies [6], and coreference resolution [33].

4 Visualization Design

In this section we discuss our visualization design that addresses our tasks, please see Fig. 2 for an overview of the encodings employed in our design.

4.1 Cluster-Word Membership

This view address (**T1**) in showing the amount of context reflected in a given cluster. Specifically, for a given word w, denote $c(w,l)$ as the number of times this word appears in the corpus with cluster $l \in [1,k]$. Then for such a cluster, we compute the percentage in which that word appears in the cluster:

$$p(w,l) = \frac{c(w,l)}{\sum_{j=1}^{k} c(w,j)}. \quad (1)$$

Thus, for cluster l we have an assigned percentage p for all words w in our corpus. We encode this as a distribution (Fig. 2(A)), where the x-axis encodes the percentage, and an area mark's height encodes how many words contain that percentage. We perform kernel density estimation to arrive at a smoothed distribution. Percentages of $p = 0$ are filtered out, as they tend to dominate, and are implicitly encoded via nonzero counts across the rest of the clusters. This view enables us to determine differences in clusters in terms of word senses. For instance, two clusters may both reflect past tense, yet they are distinguished by part-of-speech, where one cluster represents adjectives, and the other represents verbs. Our design would consequently depict overlap between these clusters (**T1**). In general, distributions that are concentrated at a value of 1 indicate only one meaning, independent of context, whereas a more even distribution across percentages indicates the dependence on context for the meaning of individual words.

4.2 Cluster Spans

This view addresses (**T2**) in showing the ability of a cluster to represent contiguous text spans. Specifically, for a given cluster, for each sentence in our corpus we group words that (a) form a contiguous span and (b) all belong to this particular cluster. We then count how many times, for a given span length, these cluster-specific spans occur over the entire corpus. We visually encode this as a heatmap (Fig. 2(B)) where each square represents a particular span length, beginning at a span of 1 (individual word), and increasing from left-to-right. We use a sequential, luminance-decreasing color map to encode count, e.g. how many times a cluster-specific span occurs in the corpus. Aligned columns of the heatmap permit a rapid comparison of span length frequencies between clusters, while a given row depicts a cluster's distribution of span frequencies. As shown in Fig. 1(B) for the last layer of the Transformer [34] model, this design enables the user to quickly assess whether certain clusters result in long spans compared to other clusters, indicative of certain types of linguistic features, e.g. named entities or a part of a constituency parse tree. This grouping of words into contiguous, cluster-specific spans is carried over to other elements of the design, namely cluster co-occurrences, as well as the detailed sentence view. Herein we refer to these grouped words as *phrases* for full generality.

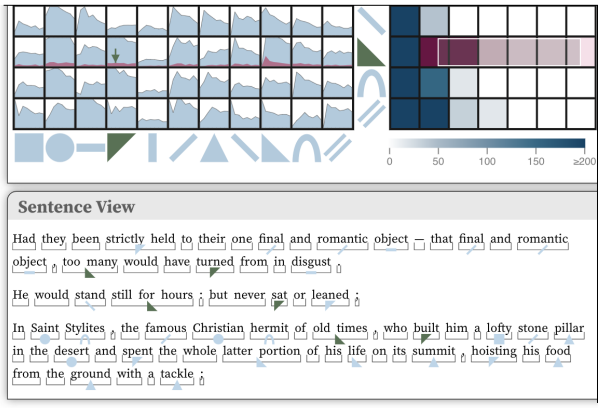

Figure 3: Here we show how the user can discover multi-word phrases of the concept of time through our interface. Brushing spans of length greater than 1, and selecting in the co-occurrence view, we obtain detailed inspections in the sentence view that enables this discovery.

4.3 Pairwise Cluster Co-occurrences

This view addresses (**T3**) in depicting relationships between clusters. More specifically, for a given phrase corresponding to a cluster, we count how many times it co-occurs with a different cluster's phrase within a given sentence. We measure co-occurrences over different spacings of phrases, e.g. phrases belonging to two different clusters might be right next to each other, but other times they might be separated by several phrases. We show these relationships in a small-multiples view of area marks: rows correspond to clusters in the first position (e.g. the left portion of the co-occurrence), while columns correspond to clusters in the second position (e.g. the right portion). The height of the area mark encodes the number of co-occurrences, while the x-axis within each cell encodes the amount of spacing between phrases, increasing from left-to-right (Fig. 2(C)). Area marks allow the user to quickly identify patterns with respect to cluster pairings. A large spike within the area mark indicates a frequent co-occurrence between clusters at a given amount of spacing, distinguished from other spacings between these clusters. This potentially indicates a salient relationship between clusters (**T3**), e.g. co-reference resolution for diagonal cells (identical clusters) or dependency relations between distinct parts-of-speech spaced a fixed amount apart. Note in Fig. 1, there are zero counts for co-occurrences that are directly next to each other in cells on the diagonal, due to the grouping of words into phrases.

Further, to visually align the different views, we associate a unique glyph with each cluster, distributed as horizontal and vertical spans within the co-occurrence view. In particular, the vertical strip of glyphs is in alignment with the rows of the span heatmap and cluster-word membership views, for quick identification of clusters amongst all views. This glyph design was chosen to handle a potentially large number of clusters. Other visual channels, e.g. color, can lead to discriminability issues, particularly for complex spatial arrangements [11]. This is characteristic of our sentence view, discussed next.

4.4 Interactions and Detailed Sentence Inspection

We allow for user interactions to (a) understand relationships between the different views, and (b) provide for detailed inspection of sentences. Specifically, the user can brush the cluster-word membership view to select words within the particular percentage range. The remaining cluster-word distributions are updated for the brushed

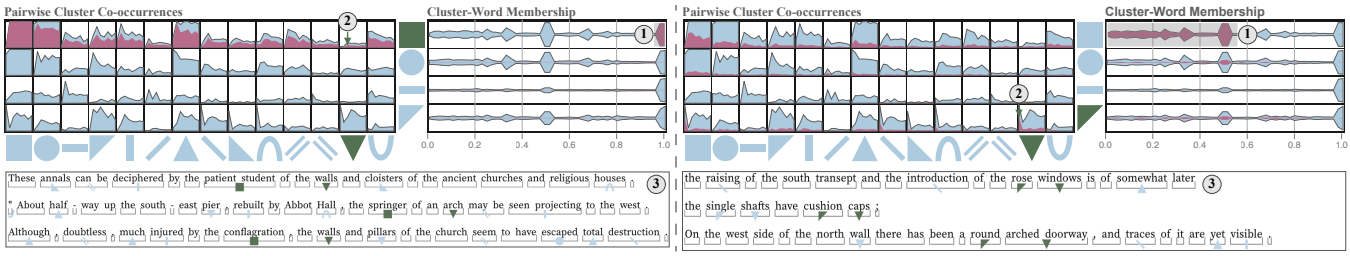

Figure 4: We show a use of our system for exploring multiple word senses, allowing the user to discover nouns that are largely context-independent (**left**), in contrast to context-dependent words shared by a different cluster that capture adjectives (**right**).

set of words with a superimposed purple area mark, in order to show more detailed relationships between clusters, shown in Fig. 1(C). Furthermore, the co-occurrence view is also updated, where we show a purple area mark for co-occurrences that contain the brushed words. We limit this filtering only to the first item (left position) of a co-occurrence. This linked update allows the user to inspect co-occurrences that have varying levels of context, depending on the user's selection. We, similarly, allow the user to brush the cluster span view, limiting phrases to the particular span lengths brushed by the user. We, further, update the co-occurrences view to this filtered set of (left positional) phrases, but limit the selection to *only* the particular cluster, in contrast with the word-cluster membership selection which impacts *all* clusters.

We also allow for the user to select both an *individual cell* and *phrase spacing* within the co-occurrence view, as indicated by the dark green arrow in Fig. 1(A), and corresponding highlighted cluster glyphs. If a user has previously performed a brushed from the aforementioned interactions, then this selection is limited to the brush query: this is shown in Fig. 1(A) by the arrow positioned on the purple area mark. For a given selection, we populate a more detailed sentence view in Fig. 1(D), where we show sentences that contain the particular pair of clusters, and spacing between clusters. The cluster-specific glyphs are carried over to this view, as well as the depiction of phrases via brackets that highlight cluster-specific contiguous spans. In Fig. 1(D), we see that the user's selection resulted in, predominantly, adjectives in the left cluster, yet these are words that can have different senses (e.g. "master" can be an adjective or noun), which arise from the user's brush of words that belong to different clusters, and are thus more context-dependent. Likewise, Fig. 3 shows an example of filtering phrases to within a certain length.

We allow the user to control various aspects of the design. They may select any of the layers within the Transformer model to load in the main view, providing a quick comparison of how contextual particular layers are – including the first layer, which is largely dependent on word embeddings and thus mostly free of context. The user can also control how many clusters to show in the visualization to reduce visual complexity, where we prioritize clusters based on the number of unique words that each cluster contains. Further, for the sentence view the user can opt to exclude glyphs of clusters not selected in the co-occurrence view, freeing clutter.

5 Results

To demonstrate our interface, we first show a use case of our system. Our interface supports the loading of an arbitrary set of sentences, but for evaluation purposes, we limit this to sentences from a book, namely "Scottish Cathedrals and Abbeys." [1] Our use case is based on this corpus, showing results for the 9'th layer of the Transformer model, please see Fig. 4. On the left side, the user first selects words that have high membership with the square cluster (1), thus limiting our view to context-independent words. The selection prompts an update to the co-occurrence view via the purple area marks representing those words, where upon clicking a pair of clusters (2) we see that the square cluster for this selection reflects nouns (3). On the right, the user next selects a range of word-cluster memberships from the same (square) cluster (1) prompting linked highlighting across clusters, thus reflective of context-dependent words/phrases. We can observe a spike in co-occurrences for this selection with respect to a pair of clusters (2), indicative of words that belong to different clusters that are right next to one another. Upon closer inspection (3), we find that this represents adjective-noun pairs, where the words classified as adjectives may also be treated as nouns, demonstrating their reliance on context.

In addition, we have gathered feedback from users, in order to assess what features participants could find by interacting with the visualization. More specifically, we conducted experiments with three graduate students, all in Computer Science, who all have some amount of experience using visual interfaces. We did not constrain them in their interactions, instead promoting free-form exploration, asking them: (1) What insights did you find by using the interface? (2) Did you find the interface easy to use?

All in all, participants found different aspects of language through the interface: one participant was able to quickly identify parts-of-speech (adjectives, nouns), while another participant found named entities in the form of dates, as well as more semantic groupings, e.g. different aspects of religion such as church, chapel, etc.. and building structures such as monument, exterior, etc.. Another participant was able to discover patterns with respect to different layers of the Transformer model, namely how the cluster-word membership becomes less unique in later layers, as well as more diverse span lengths. Participants, however, did find the design to be rather complex. One participant mentioned that it took some time to understand it, but afterwards, they were able to navigate amongst the views. Another participant, however, found the complexity to be too overwhelming at times, which inhibited their discovery.

6 Conclusion

We introduced a method for visually analyzing contextualized embeddings produced from deep, pre-trained, language models. Our visualization design takes inspiration from, and abstracts, the class of supervised probes traditionally used to interpret language models, in order to enable a more general analysis of contextualized embeddings. We find preliminary user feedback to be encouraging, however, in the future we plan on obtaining feedback from domain experts within NLP as well as linguistics to assess the design's effectiveness. Furthermore, we plan on extending our work to enable a more comparative analysis of contextualized embeddings, particularly across layers, in order to understand what linguistic properties are learned amongst different representations.

[1] texts from books are acquired from Project Gutenberg (`https://www.gutenberg.org/`). Though the main results shown are based on only one book, Fig. 3 shows our interface for "Moby Dick".

REFERENCES

[1] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, p. 1027–1035, 2007.

[2] Y. Belinkov and J. Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, 2019.

[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[4] J. Bjerva, B. Plank, and J. Bos. Semantic tagging with deep residual networks. *arXiv preprint arXiv:1609.07053*, 2016.

[5] A. Boggust, B. Carter, and A. Satyanarayan. Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples. *arXiv preprint arXiv:1912.04853*, 2019.

[6] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. 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*, pp. 276–286, 2019.

[7] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni. 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)*, pp. 2126–2136, 2018.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] M. Gardner, Y. Artzi, V. Basmova, J. Berant, B. Bogin, S. Chen, P. Dasigi, D. Dua, Y. Elazar, A. Gottumukkala, et al. Evaluating nlp models via contrast sets. *arXiv preprint arXiv:2004.02709*, 2020.

[10] S. Gehrmann, H. Strobelt, R. Krüger, H. Pfister, and A. M. Rush. Visual interaction with deep learning models through collaborative semantic inference. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):884–894, 2019.

[11] S. Haroz and D. Whitney. How capacity limits of attention influence information visualization effectiveness. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2402–2410, 2012.

[12] F. Heimerl and M. Gleicher. Interactive analysis of word vector embeddings. In *Computer Graphics Forum*, vol. 37, pp. 253–265. Wiley Online Library, 2018.

[13] J. Hewitt and P. Liang. 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)*, pp. 2733–2743, 2019.

[14] J. Hewitt and C. 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)*, pp. 4129–4138, 2019.

[15] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE transactions on visualization and computer graphics*, 25(8):2674–2693, 2018.

[16] G. Jawahar, B. Sagot, and D. Seddah. What does bert learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3651–3657, 2019.

[17] T. Kim, J. Choi, D. Edmiston, and S.-g. Lee. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. In *International Conference on Learning Representations*, 2019.

[18] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith. Linguistic knowledge and transferability of contextual 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)*, pp. 1073–1094, 2019.

[19] S. Liu, P.-T. Bremer, J. J. Thiagarajan, V. Srikumar, B. Wang, Y. Livnat, and V. Pascucci. Visual exploration of semantic relationships in neural word embeddings. *IEEE transactions on visualization and computer graphics*, 24(1):553–562, 2017.

[20] S. Liu, Z. Li, T. Li, V. Srikumar, V. Pascucci, and P.-T. Bremer. Nlize: A perturbation-driven visual interrogation tool for analyzing and interpreting natural language inference models. *IEEE transactions on visualization and computer graphics*, 25(1):651–660, 2018.

[21] Y. Liu, E. Jun, Q. Li, and J. Heer. Latent space cartography: Visual analysis of vector space embeddings. In *Computer Graphics Forum*, vol. 38, pp. 67–78. Wiley Online Library, 2019.

[22] Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, and H. Qu. Understanding hidden memories of recurrent neural networks. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 13–24. IEEE, 2017.

[23] C. Park, I. Na, Y. Jo, S. Shin, J. Yoo, B. C. Kwon, J. Zhao, H. Noh, Y. Lee, and J. Choo. Sanvis: Visual analytics for understanding self-attention networks. In *2019 IEEE Visualization Conference (VIS)*, pp. 146–150. IEEE, 2019.

[24] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 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)*, pp. 2227–2237, 2018.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners.

[26] A. Ravichander, Y. Belinkov, and E. Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? *arXiv preprint arXiv:2005.00719*, 2020.

[27] E. Reif, A. Yuan, M. Wattenberg, F. B. Viegas, A. Coenen, A. Pearce, and B. Kim. Visualizing and measuring the geometry of bert. In *Advances in Neural Information Processing Systems*, pp. 8594–8603, 2019.

[28] N. Rethmeier, V. K. Saxena, and I. Augenstein. Tx-ray: Quantifying and explaining model-knowledge transfer in (un-) supervised nlp. *arXiv preprint arXiv:1912.00982*, 2019.

[29] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912. Association for Computational Linguistics, Online, July 2020.

[30] T. Spinner, U. Schlegel, H. Schäfer, and M. El-Assady. explainer: A visual analytics framework for interactive and explainable machine learning. *IEEE transactions on visualization and computer graphics*, 26(1):1064–1074, 2019.

[31] H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush. S eq 2s eq-v is: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics*, 25(1):353–363, 2018.

[32] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE transactions on visualization and computer graphics*, 24(1):667–676, 2017.

[33] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. V. Durme, S. Bowman, D. Das, and E. Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[35] J. Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42, 2019.

[36] U. Von Luxburg. *Clustering stability: an overview*. Now Publishers Inc, 2010.

[37] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.