NxPlain: A Web-based Tool for Discovery of Latent Concepts

Fahim Dalvi Nadir Durrani Hassan Sajjad[♣*]
Tamim Jaban Mus'ab Husaini Ummar Abbas
{faimaduddin,ndurrani}@hbku.edu.qa
Qatar Computing Research Institute, HBKU Research Complex, Doha, Qatar
[♣]Faculty of Computer Science, Dalhousie University, Halifax, Canada

Abstract

The proliferation of deep neural networks in various domains has seen an increased need for the interpretability of these models, especially in scenarios where fairness and trust are as important as model performance. A lot of independent work is being carried out to: i) analyze what linguistic and non-linguistic knowledge is learned within these models, and ii) highlight the salient parts of the input. We present **NxPlain**, a web application that provides an explanation of a model's prediction using latent concepts. NxPlain discovers latent concepts learned in a deep NLP model, provides an interpretation of the knowledge learned in the model, and explains its predictions based on the used concepts. The application allows users to browse through the latent concepts in an intuitive order, letting them efficiently scan through the most salient concepts with a global corpus-level view and a local sentence-level view. Our tool is useful for debugging, unraveling model bias, and for highlighting spurious correlations in a model. A hosted demo is available here: https://nxplain.qcri.org[1]

1 Introduction

Interpretation of deep neural networks (DNNs) has gained a lot of attention in recent years, especially in NLP, where state-of-the-art models are being widely deployed and used in practice. Work done in interpretation can be broadly classified into two branches: i) representation analysis and ii) attribution analysis. The former attempts to understand what knowledge is learned within the representation (Belinkov et al., 2017a; Tenney et al., 2019) and the latter is focused on how the model predicts the output (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018).[2]

A drawback of the methods in *representation analysis* is that it does not gauge whether the model uses what it has learned in making a prediction. On the other hand, the drawback of *attribution analysis* is that their explanations are limited to discrete units (e.g. words, some specific piece of the network), and the abstract nuances behind these discrete units are lost in the explanation, resulting in an inadequate or implausible explanation. Some efforts have been made in trying to connect representation and attribution analysis (Feder et al., 2021; Elazar et al., 2021).

In this work, we present **NxPlain**, a web-app that provides a holistic view by combining representation and attribution analysis. More specifically, we discover latent concepts in the model using the Latent Concept Analysis (Dalvi et al., 2022) and connect these concepts to specific predictions using Integrated Gradients (Sundararajan et al., 2017), a model and input saliency method.

NxPlain allows the users to:

- Discover latent concepts in *transformers* (Wolf et al., 2020) models via an interactive GUI
- Align the concepts using human-defined ontologies and task specific concepts
- Explain predictions using saliency-based attributions and extracted latent concepts

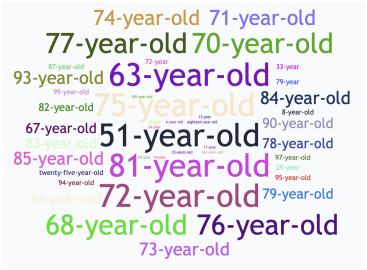
The analysis presented by **NxPlain** can enable a practitioner to understand a trained model better and be aware of the kinds of concepts a model is using to perform its tasks. For example, the word *immigrant* can appear as part of a neutral concept (if the model clusters it with other "roles" related to a person's status like "non-immigrant", "resident",

[*] This work was carried out while the author was at QCRI.
[1] A short video demo of the system is also available here: https://www.youtube.com/watch?v=C2Pi04fI5dk
[2] The following survey papers summarize the work done on *Representations Analysis* (Belinkov et al., 2020; Sajjad et al., 2021) and *Attribution Analysis* (Danilevsky et al., 2020)

(a) Terms used in hate-speech against immigration policies

(b) Syntactic concept of hyphenated words

(c) Concept made up of numbers



(d) Lexical concept (hyphenation) representing ages

(e) Morphological concept (*adjectives* with common suffix *est* signifying superlative adjectives

(f) Named entities in Germany

Figure 1: Examples of Latent Concepts.

etc), or it can appear as part of a negative concept (if the model clusters it with other hate-speech related terms like "alien", "illegal" etc.) as in Figure 1. Understanding which of these categorizations a model is learning and relying on can be a strong signal of the underlying biases of the model. A more benign example of debugging would also be able to see a purely lexical concept being used for prediction (say words ending in "y"), when the lexical property should not have any bearing on the task at hand. The target users for our system can be broadly divided into two categories: i) researchers/practitioners who want to understand their model better, and ii) other systems that want to use the concepts extracted by NxPlain to better explain predictions to their customers.

2 System Design

The overall system behind the **NxPlain** application is split into three distinct components. See Figure 2 for a pictorial representation.

- **Backend**: This part of the app integrates the `pipeline`, which handles i) extraction of latent concepts, ii) computation of various orderings, and iii) computation of the concepts relevant to particular sentences etc. A database is used to store all of the computed results so that the other two components can then use these results.
- **`Rest API`**: This piece displays the results from the `Backend` in an organized and machine-readable fashion. Users can use this to access the latent concepts and their relevant metadata for their applications.
- **`Frontend`**: This is the primary user-facing module of the app, and runs in a Web browser. The `frontend` provides an easy to use the graphical interface to add models to the computation queue and retrieve the extracted concepts once they are ready. Figure 4 shows the *Model Explanations* page, where one can browse all the extracted concepts, sort them according to various criteria and analyze the knowledge learned in the selected model.

Technical Details For extracting the concepts, we use the code provided by Dalvi et al. (2022). We then tag the input corpus with various human-defined tagsets such as Parts-of-Speech and Semantic tags, and align the latent concepts with these, as done by Sajjad et al. (2022). The results are then stored in a database, and retrieved later via a Python server implemented using Flask. The backend exposes a Rest API which can be used as-is by users in their own applications. We also provide an Angular frontend app that uses the Rest API to present the concepts in a GUI. For sentence-level explanations, we use the (Kokhlikyan et al.,

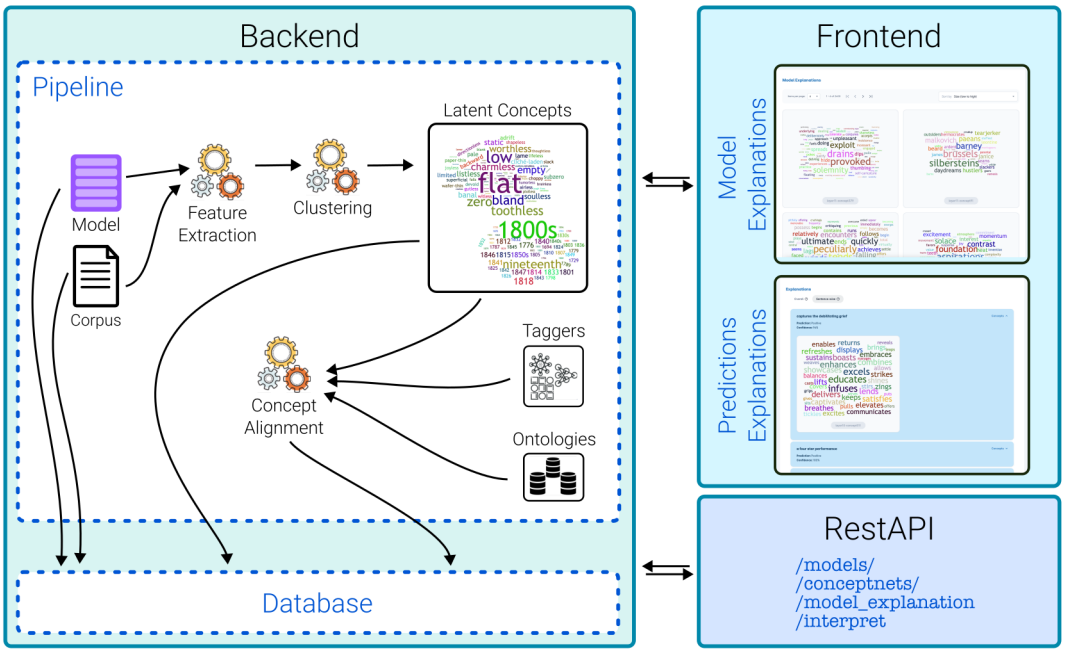


Figure 2: The architecture of NxPlain: The backend uses a pipeline to extract latent concepts and align them with various human ontologies and task-specific concepts. The frontend then uses the computed data to provide both global (model-level) and local (prediction-level) explanations. A RestAPI is also provided so a user can build upon the backend without having to use the provided frontend.

2020) tookit's Integrated Gradients implementation to perform attribution analysis.

3 Pipeline Components

The **NxPlain** application provides an easy interface to analyze the latent knowledge learned within a deep NLP model, as well as connect these latent concepts to specific predictions. In order to do this, the pipeline in the Backend relies on three key components proposed by recent literature: i) concept discovery, ii) concept alignment, and iii) attribution analysis.

3.1 Concept Discovery

The first component, responsible for extracting the latent concepts learned by a model is based on work done by Dalvi et al. (2022), called *Latent Concept Analysis*. At a high level, feature vectors (contextualized representations) are first generated by performing a forward pass on the model. These representations are then clustered using agglomerative hierarchical clustering (Gowda and Krishna, 1978) to discover the encoded concepts. The hypothesis is that contextualized word representations learned within pretrained language models capture *meaningful* groupings based on a coherent concept such as lexical, syntactic and semantic similarity, or any task or data specific pattern that groups the words together (Dalvi et al., 2022). Figure 1 shows example concepts discovered in the model space of a base and finetuned BERT model. The concepts discovered are a mix of linguistic, lexical and semantic concepts.

3.2 Concept Alignment

The second component uses an alignment framework proposed by Sajjad et al. (2022) to align each of the latent concepts to some pre-existing ontology like part-of-speech, semantic tags, WordNet etc. This enables richer explanations for the latent concepts, and also allows for the application to sort all of the concepts based on criteria relevant to the user. For instance, if the user is only interested in morphological latent concepts, the application can easily filter and sort all of the latent concepts based on this property after the alignment has been performed.

The alignment of a concept to a specific property (e.g. Noun) is done by checking if most of the words (above a certain threshold) in the concept are labeled with that property. For example, $C_{pos}(JJR) =$

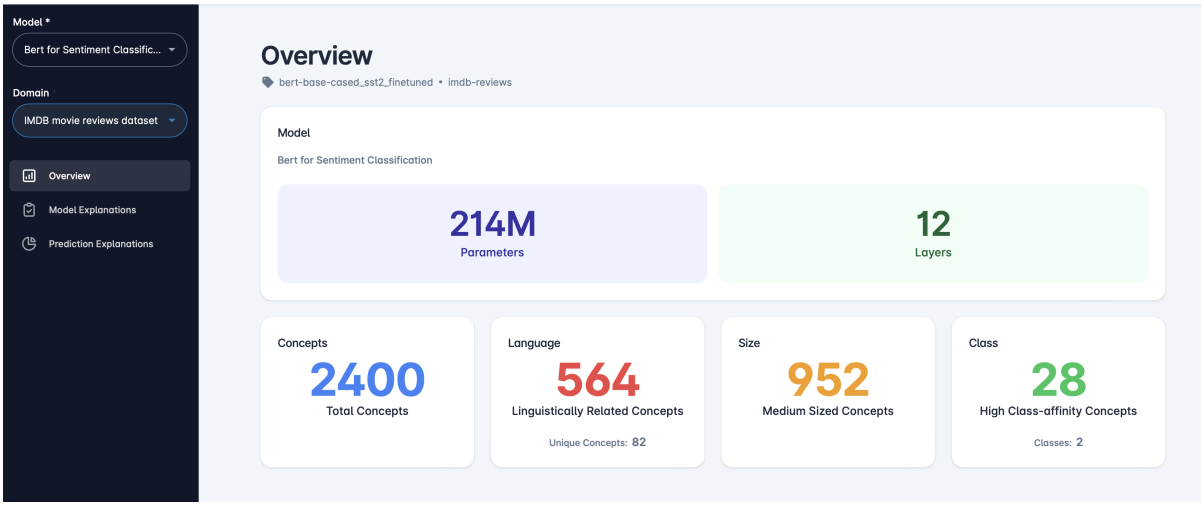


Figure 3: Sample overview page, providing high level statistics at a glance.

$\{greener, taller, happier, \ldots\}$ would be aligned to the property of "comparative adjectives" in the POS tagging task, $C_{sem}(MOY) = \{January, February, \ldots, December\}$ defines a concept containing months of the year in the semantic tagging task, and $C_{muslim}(names) = \{Ahmed, Muhammad, Karim, Hamdy, \ldots\}$ represents a concept of Muslim names. Explanations based on human-defined concepts are not always applicable or available as these models learn very fine-grained hierarchies of knowledge and concepts that are very task-specific, hence not every latent concept is aligned to some pre-existing tag/ontology.

3.3 Attribution Analysis

Our first two components are geared towards understanding what the model has learned, however, it does not necessarily imply that this knowledge is utilized during prediction and provides no insight into how these concepts are being used. To bridge this gap, our third component uses **Integrated Gradients** (IG) (Sundararajan et al., 2017), which is a powerful axiomatic attribution method for deep neural networks that computes the importance of input features and model components based on their contribution to model's prediction. More concretely, IG is used to extract the salient input features (words) used to make a certain prediction, and these salient features are then mapped to latent concepts to expand on the explanation. For example in Figure 5 highlights "captures" to be the most salient input feature used in predicting the sentiment of the sentence.

4 Frontend Views

The goal of **NxPlain** is to provide an easy method for users to extract and analyze latent knowledge learned within a deep NLP model and connect them to the prediction. The `Frontend` helps achieve this goal by providing a intuitive yet powerful GUI that can be used to interact with a model's latent concepts and predictions. The user can upload a model and a corpus that they want to analyze. The computational queue of the application discovers latent concepts and aligns them using the components mentioned in Section 3. The user can then use the `Frontend`, where they can switch between three major views:

Overall view: This view presents a high-level overview of the concepts learned by the model. Specifically, we can see i) the number of concepts learned, ii) statistics on the concepts aligned with the human-fined concepts, iii) a summary of the size distribution of these concepts, iv) and salient concepts in the data and model. Figure 3 shows a sample overview page for a Sentiment analysis model.

Model Explanations view: This view presents the latent concepts in a paginated view, along with controls to sort the concepts. Users can sort the concepts i) by size, ii) by their affinity to the linguistic phenomenon (using the alignments computed earlier), iii) by their relation to the various output classes (in classification models) and iv) by their overall relevance. Each concept is accompanied by a unique label to keep track of important concepts. See Figure 4 for a sample model explanation view.

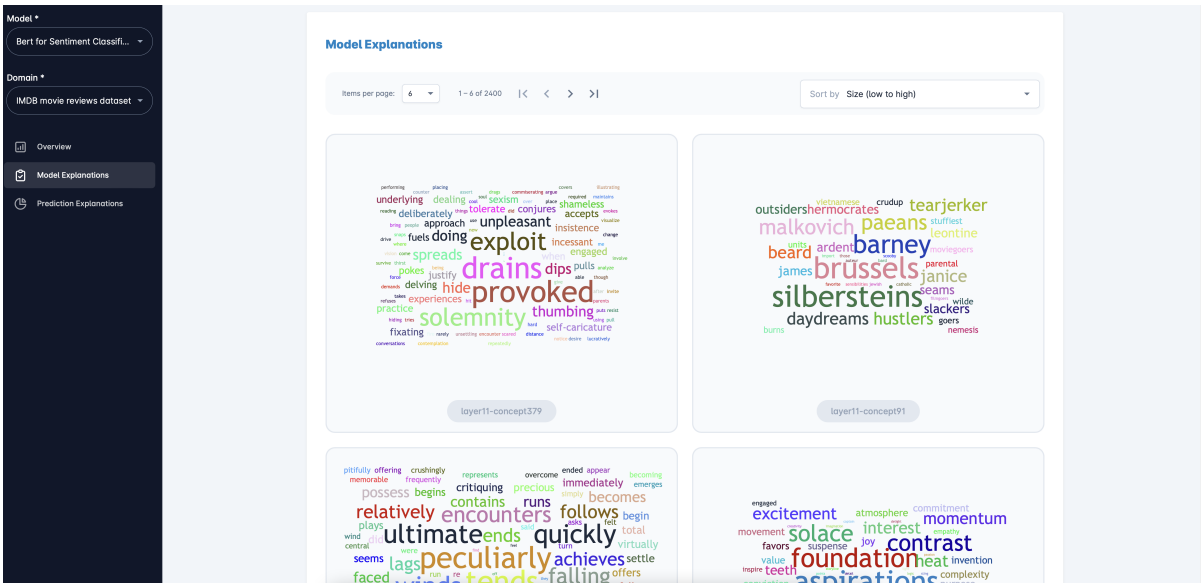


Figure 4: The model-explanation page showing latent concepts for the selected model and domain. Sorting and pagination controls allow a user to effectively browse and analyze concepts learned by the model.

Prediction Explanations view: This view allows the user to look at concepts used in making a prediction and facilitates a deeper view of the behavior of the model on specific sentences. The *attribution analysis* component is used to get a salience map of the input tokens, as well as the matching concepts that contain these tokens in similar contexts. Figure 5 displays the prediction view, where the user can select the sentences that they want to analyze. Here NxPlain shows that "captures" was the most influential word used by the model to make the prediction. The model used a latent concept representing *positive verbs* to make the prediction.

5 Related Work

5.1 Toolkits

A number of toolkits have been made available to carry out analysis of neural network models. Google's What-If tool (Wexler et al., 2019) inspects machine learning models and provides users an insight into the trained model based on the predictions. Seq2Seq-Vis (Strobelt et al., 2018) enables the user to trace back the prediction decisions to the input in NMT models. Captum (Kokhlikyan et al., 2020) provides generic implementations of a number of gradient and perturbation-based attribution algorithms. NeuroX (Dalvi et al., 2019) and Ecco (Alammar, 2021) use probing classifiers to examine the representations pre-trained language models. ConceptX (Alam et al., 2023) provides a framework for analyzing and annotating latent concepts in pre-trained language models. Tenney et al. (2020) facilitates debugging of pLMs through interactive visualizations. Our work is different from these toolkits. Our toolkit bridges the gap between representation analysis and causation by using attribution-based method. NxPlain provides enriched explanations using traditional linguistic knowledge and human-defined ontologies.

5.2 Research Works

A large number of studies primarily focus on understanding the knowledge learned within a trained model. Researchers have proposed numerous analysis frameworks such as diagnostic classifiers (Belinkov et al., 2017a; Hupkes et al., 2018), corpus analysis (Kádár et al., 2017; Poerner et al., 2018; Na et al., 2019), linguistic correlation analysis (Lakretz et al., 2019; Durrani et al., 2022a). A plethora of work has been carried out using these analyses frameworks to analyze what concepts are learned within the representations through relevant extrinsic phenomenon varying from word morphology (Vylomova et al., 2017; Belinkov et al., 2017a; Dalvi et al., 2017) to high level concepts such as structure (Shi et al., 2016; Linzen et al., 2016; Durrani et al., 2019) and semantics (Qian et al., 2016; Belinkov et al., 2017b; Durrani et al., 2021) or more generic properties such as sentence length (Adi et al., 2016; Bau et al., 2019).

While the work done on representation analysis unwraps interesting insights about the knowledge

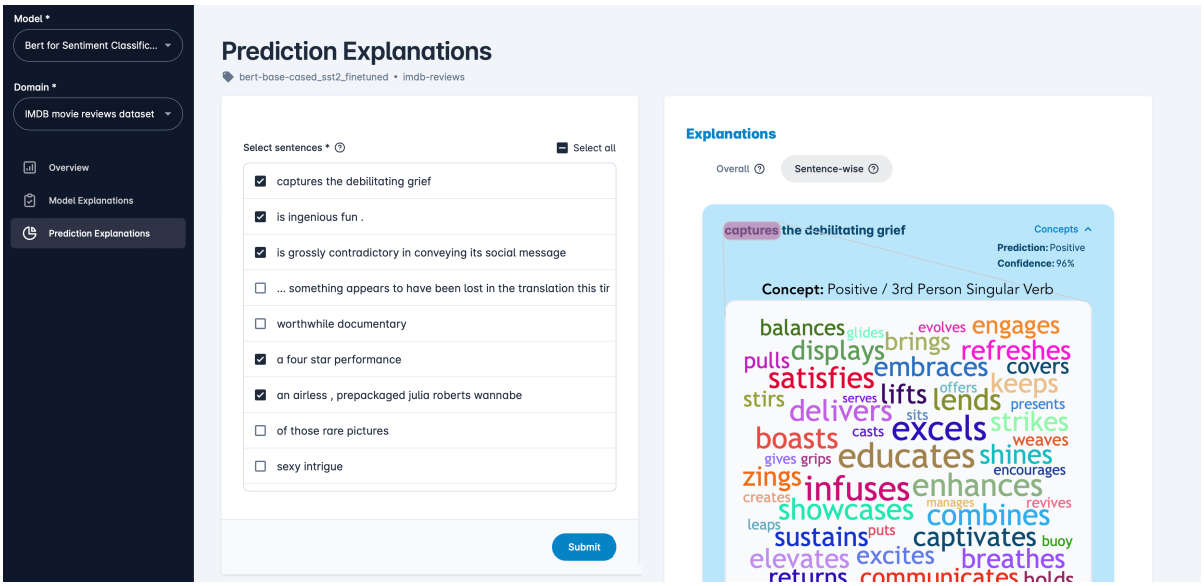


Figure 5: The prediction-explanation page showing latent concepts used during the prediction. The Integrated Gradients method highlights that capture is the most salient word used in the prediction. NxPlain connects it to the concept used along with its label. We observe here that the model used a concept representing positive verbs.

learned within the network and how it is preserved, it's only limited to human-defined concepts. More recent work has discovered that these models capture novel ontologies (Michael et al., 2020; Dalvi et al., 2022; Fu and Lapata, 2022) learning linguistic concepts (Sajjad et al., 2022), as well as the task-specific concepts (Durrani et al., 2022b) that emerge as the pre-trained language models are fine-tuned towards a task.

Another line of work in interpretability focuses on attribution analysis that characterizes the role of model components and input features towards a specific prediction (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018). The explanations are categorized based on two aspects: local or global (Guidotti et al., 2018). The former gives a view of explanation at a level of individual instance (Ribeiro et al., 2016; Alvarez-Melis and Jaakkola, 2017), whereas the latter explains the general behavior of the model at corpus level (Pryzant et al., 2018; Pröllochs et al., 2019).

6 Conclusion

We presented **NxPlain**, a web-app for connecting concept analysis with model prediction. The application bridges *representation analysis* and *attribution analysis* to better explain the models' predictions, and provides a intuitive, yet powerful graphical interface to explore the knowledge learned by a model, and also to pinpoint the knowledge used in specific predictions. In the future, we plan to enable human-in-the-loop to enhance concept alignment, as well as incorporate feedback into the explanation system. A hosted version of the application can be accessed at `https://nxplain.qcri.org`.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. *arXiv preprint arXiv:1608.04207*.

Firoj Alam, Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Abdul Rafae Khan, and Jia Xu. 2023. Conceptx: A framework for latent concept analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*.

J Alammar. 2021. Ecco: an open source library for the explainability of transformer language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 249–257.

David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark. Association for Computational Linguistics.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *Proceedings of the Seventh*

International Conference on Learning Representations, ICLR '19, New Orleans, USA.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2020. On the linguistic representational power of neural machine translation models. *Computational Linguistics*, 45(1):1–57.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, IJCNLP '17, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. 2017. Understanding and Improving Morphological Learning in the Neural Machine Translation Decoder. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in BERT. In *Proceedings of the Tenth International Conference on Learning Representations*, ICLR '22, Online.

Fahim Dalvi, Avery Nortonsmith, D. Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, and James Glass. 2019. Neurox: A toolkit for analyzing individual neurons in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI '19, pages 9851–9852, Honolulu, USA.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Nadir Durrani, Fahim Dalvi, and Hassan Sajjad. 2022a. Linguistic correlation analysis: Discovering salient neurons in deepnlp models.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL '19, pages 1504–1516, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, and Fahim Dalvi. 2021. How transfer learning impacts linguistic knowledge in deep NLP models? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4947–4957, Online. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Firoj Alam. 2022b. On the transformation of latent space in fine-tuned nlp models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, EMNLP, Abu Dhabi, UAE. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386.

Yao Fu and Mirella Lapata. 2022. Latent topology induction for understanding contextualized representations.

K Chidananda Gowda and G Krishna. 1978. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 10(2):105–112.

Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. 2018. A survey of methods for explaining black box models. *CoRR*, abs/1802.01933.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL '18, pages 1195–1205, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *arXiv:1711.10203*.

Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 43(4):761–780.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521– 535.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Julian Michael, Jan A. Botha, and Ian Tenney. 2020. Asking without telling: Exploring latent ontologies in contextual representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 6792–6812, Online. Association for Computational Linguistics.

Seil Na, Yo Joong Choe, Dong-Hyun Lee, and Gunhee Kim. 2019. Discovery of natural language concepts in individual units of CNNs. *CoRR*, abs/1902.07249.

Nina Poerner, Benjamin Roth, and Hinrich Schütze. 2018. Interpretable textual neuron representations for NLP. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 325–327, Brussels, Belgium. Association for Computational Linguistics.

Nicolas Pröllochs, Stefan Feuerriegel, and Dirk Neumann. 2019. Learning interpretable negation rules via weak supervision at document level: A reinforcement learning approach. 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 407–413, Minneapolis, Minnesota. Association for Computational Linguistics.

Reid Pryzant, Sugato Basu, and Kazoo Sone. 2018. Interpretable neural architectures for attributing an ad's performance to its writing style. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 125–135, Brussels, Belgium. Association for Computational Linguistics.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. Investigating Language Universal and Specific Properties in Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, pages 1478–1488, Berlin, Germany. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2021. Neuron-level Interpretation of Deep NLP Models: A Survey. *CoRR*, abs/2108.13138.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. 2022. Analyzing encoded concepts in transformer language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3082–3101, Seattle, United States. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, EMNLP '16, pages 1526–1534, Austin, TX, USA.

Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander Rush. 2018. Debugging sequence-to-sequence models with Seq2Seq-vis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 368–370, Brussels, Belgium. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365.

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.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 107–118, Online. Association for Computational Linguistics.

Ekaterina Vylomova, Trevor Cohn, Xuanli He, and Gholamreza Haffari. 2017. Word representation models for morphologically rich languages in neural machine translation. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 103–108, Copenhagen, Denmark. Association for Computational Linguistics.

James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viegas, and Jimbo Wilson. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.