femington

Int. J. Intell. Syst. Data Sci. (2026) 1(3):1–10
Article ID: IJISDS-2026-275

Perturbation-Driven Visual Analytics for Probing Internal Representations in Neural Language Models

Eeshwar Pasula
eshwarpasula@gmail.com
Independent Researcher

Abstract

This paper introduces an interactive visual analytics framework designed to support exploratory analysis of neural network models for natural language inference through a perturbation-driven paradigm. Rather than treating trained models as static black boxes, the system enables users to dynamically manipulate inputs, internal attention mechanisms, and output predictions while observing corresponding changes across the processing pipeline. The interface integrates multiple coordinated views including bipartite graph and matrix representations of attention, barycentric coordinate plots for probabilistic predictions, and a pipeline visualization for tracking parameter updates to support hypothesis formation and causal reasoning about model behavior. A constrained optimization procedure, inspired by the margin-infused relaxed algorithm, allows users to correct erroneous predictions while minimizing parameter deviation and to compare the relative influence of encoder, attention, and classifier stages on prediction outcomes. The system also overlays syntactic dependency structures onto attention visualizations, enabling grammar-guided simplification and facilitating investigation of the relationship between linguistic structure and learned alignments. Evaluation with NLP researchers demonstrates that the tool supports a range of analytical tasks, including stability assessment, error diagnosis, attention editing, and comparative analysis of model components. The framework is implemented as a lightweight Python library that integrates with existing PyTorch models, lowering the barrier to adoption for routine model interrogation.

Keywords

• Visual Analytics • Neural Language Models • Perturbation • Attention Visualization • Interpretability • Natural Language Inference • Constrained Optimization

1. Introduction

Deep neural networks have become the dominant approach for natural language processing tasks, yet their internal representations remain notoriously difficult to interpret. Models for natural language inference (NLI), which determine whether a hypothesis can be inferred from a premise, often achieve high accuracy while exhibiting unexpected failure modes and biases [1]. Understanding these models is essential for diagnosing errors, ensuring reliability, and building trust in deployed systems. Traditional evaluation metrics provide aggregate performance measures but offer little insight into individual prediction behavior or the internal mechanisms that produce specific outcomes.

Recent work in interpretability has produced tools for visualizing attention weights and saliency maps [2, 3], but these typically present static views that do not support interactive exploration or causal reasoning about model behavior.

© 2026 Femington Academic Publishing. All rights reserved.

This paper introduces a perturbation-driven visual analytics framework that enables researchers to probe neural language models through direct manipulation. Rather than treating trained models as static black boxes, the system allows users to dynamically modify inputs, intervene on internal attention mechanisms, and adjust output predictions while observing corresponding changes across the processing pipeline. This interactive approach supports hypothesis formation and causal reasoning about how models represent and process linguistic information [4].

The framework integrates multiple coordinated views to provide comprehensive insight into model behavior. Bipartite graph and matrix representations visualize attention patterns between tokens, revealing which parts of the input the model focuses on during inference [5]. Barycentric coordinate plots display probabilistic predictions in a 2D space, enabling rapid assessment of prediction confidence and ambiguity. A pipeline visualization tracks parameter updates and information flow through encoder, attention, and classifier stages.

A constrained optimization procedure, inspired by the margin-infused relaxed algorithm [6], allows users to correct erroneous predictions while minimizing parameter deviation. The system also overlays syntactic dependency structures onto attention visualizations, facilitating investigation of the relationship between grammatical structure and learned alignments. Evaluation with NLP researchers demonstrates that the tool supports a range of analytical tasks, and the framework is implemented as a lightweight Python library that integrates with existing PyTorch models [7], lowering the barrier to adoption for routine model interrogation.

2. Related Work

The field of visual analytics for neural language models has drawn upon advances from multiple research areas. This section situates our perturbation-driven framework within the broader research landscape.

2.1 Attention Visualization for Neural Language Models

Early tools for probing transformer attention focused on revealing which input tokens a model attends to. Vig [5] introduced BertViz, a multi-scale attention visualization that operates at the neuron, head, and model levels; this work established the bipartite-graph and heatmap idioms that our framework extends with interactive perturbation. Park et al. [8] proposed SANVis, a visual analytics system specifically designed for multi-head self-attention networks, demonstrating how coordinated views can help analysts compare attention across heads. DeRose et al. [9] introduced Attention Flows, which allows users to trace and compare attention within and across layers of fine-tuned transformer models, directly informing our design of coordinated attention views.

More recently, Yeh et al. [10] presented AttentionViz, which embeds query and key vectors jointly to offer a global perspective on transformer attention across multiple sequences—a complementary approach to our per-example perturbation paradigm. Neuron-level exploration of large language models has also been addressed by NeuronautLLM [11], which visualizes influential neurons in relation to user-defined prompts.

2.2 Interpretability Tools for NLP

Tenney et al. [4] released the Language Interpretability Tool (LIT), an extensible open-source platform for visualization and analysis of NLP models that integrates local explanations, aggregate statistics, and

counterfactual generation. Li et al. [12] developed T3-Vis, a visual analytic framework targeted at assisting researchers during the training and fine-tuning of transformers, revealing hidden states and attention importance scores through interactive views. The question of whether attention weights constitute valid explanations has been rigorously debated: Jain and Wallace [2] showed that standard attention modules do not provide meaningful explanations, while Wiegreffe and Pinter [3] subsequently proposed alternative tests demonstrating conditions under which attention can serve as faithful explanation.

2.3 Probing, Interpretability, and Visual Analytics for XAI

Belinkov [13] provides a systematic survey of probing classifiers for linguistic structure, which motivates our use of syntactic dependency overlays to test hypotheses about what the model has learned. La Rosa et al. [14] offer a comprehensive state-of-the-art review of visual analytics for explainable deep learning, confirming that interactive perturbation remains an under-explored modality. Alicioglu and Sun [15] similarly survey visual analytics methods for XAI, highlighting the gap between static saliency maps and interactive hypothesis-testing tools. The reliability of gradient-based analysis for NLP interpretability was questioned by Wang et al. [16], who demonstrated that gradients are easily manipulable—a finding that motivates our attention-editing approach as an alternative.

2.4 Transformer Architecture and Model Components

The Transformer architecture underlying our target models was introduced by Vaswani et al. [17], whose multi-head self-attention mechanism is the primary object of our visualization and perturbation experiments. Devlin et al. [1] subsequently introduced BERT, establishing the encoder-only pre-training paradigm that dominates NLI benchmarks and serves as the primary model class targeted by our framework. Voita et al. [18] studied the functional roles of individual attention heads, finding that only a subset perform specialized linguistic operations—a finding directly exploited by our attention-head comparison capability.

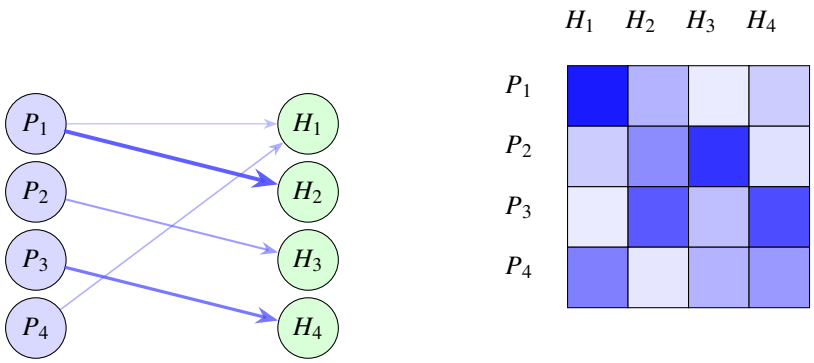
2.5 Optimization and Infrastructure

Our constrained optimization for error correction is grounded in the margin-infused relaxed algorithm (MIRA) of Crammer and Singer [6], an online multiclass learner that minimizes parameter updates subject to margin constraints. The implementation relies on PyTorch [7] for automatic differentiation, enabling gradient computation through the full model pipeline. Interactive web-based views are built on D3.js [19], an established data-driven document visualization library, with syntactic annotations provided by the Stanford CoreNLP toolkit [20].

3. Methodology

3.1 Framework Architecture

The perturbation-driven visual analytics framework consists of four primary components: a model interface that wraps PyTorch [7] models and exposes internal representations; a perturbation engine that applies user-specified modifications to inputs, attention weights, or parameters; a visualization server that generates coordinated multiple views; and an optimization module that computes minimal-perturbation corrections for erroneous predictions.



(a) Bipartite Graph **(b) Attention Matrix**

Figure 1: Complementary attention representations. (a) Bipartite graph illustrating interactions between primary nodes (P_i) and hidden nodes (H_i), where edge thickness indicates attention strength. (b) Corresponding attention matrix, with darker cells representing higher attention weights.

The model interface hooks into forward and backward passes of transformer-based language models [17], capturing attention matrices, hidden states, and gradient information without modifying the underlying computation graph. This non-invasive approach ensures that the framework can be used with any PyTorch model without requiring architectural changes.

3.2 Attention Visualization

Attention patterns are visualized using two complementary representations: bipartite graphs and matrix heatmaps, following idioms established in prior work [5, 8]. The bipartite graph view displays tokens from the premise and hypothesis as two columns of nodes, with edges representing attention weights between them. Edge thickness and opacity encode attention magnitude, enabling rapid identification of strong alignments. Users can filter edges by weight threshold and highlight paths corresponding to specific linguistic relationships.

The matrix view presents attention weights as a heatmap with premise tokens on one axis and hypothesis tokens on the other. This representation facilitates quantitative comparison of attention patterns across layers and heads, similar to the matrix-based views in Attention Flows [9]. Both views are linked, so selection in one view highlights corresponding elements.

3.3 Probabilistic Prediction Visualization

Model predictions for natural language inference typically produce probabilities for three classes: entailment, contradiction, and neutral [1]. These three probabilities sum to one, forming a 2D simplex that can be visualized using barycentric coordinates. Our framework plots each prediction as a point within an equilateral triangle, where distance to each vertex corresponds to probability of that class.

This representation enables rapid assessment of prediction confidence (points near vertices indicate high confidence, points near the center indicate uncertainty) and facilitates comparison of predictions across multiple inputs or model variants. Users can hover over points to inspect the corresponding input text and attention patterns.

3.4 Perturbation Engine

The perturbation engine enables three types of interventions:

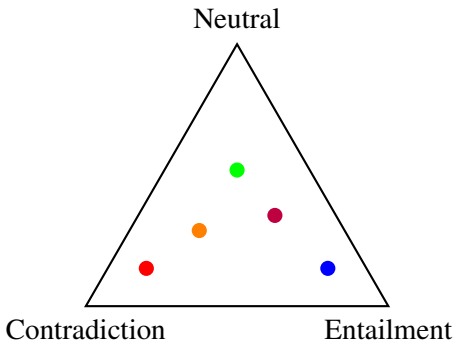


Figure 2: Barycentric coordinate visualization of three-class predictions (entailment, contradiction, neutral).

Input perturbation: Users can modify individual tokens in the premise or hypothesis and observe how attention patterns and predictions change. Token modifications include substitution, insertion, deletion, and masking. The system highlights tokens whose modification produces the largest change in attention or prediction. The reliability of such gradient-based token importance signals has been studied by Wang et al. [16], motivating our attention-editing alternative.

Attention editing: Users can directly manipulate attention weights by strengthening or weakening specific connections in the bipartite graph view. The system recomputes the forward pass with modified attention weights, enabling exploration of causal relationships between attention patterns and predictions. This design is informed by the debate on whether attention constitutes an explanation [2, 3].

Parameter perturbation: Users can apply noise to encoder, attention, or classifier parameters and observe the effect on model behavior. This supports stability assessment and sensitivity analysis.

3.5 Constrained Optimization for Error Correction

Inspired by the margin-infused relaxed algorithm (MIRA) [6], we implement a constrained optimization procedure that corrects erroneous predictions while minimizing parameter deviation. Given a misclassified example, the optimization finds minimal adjustments to model parameters such that the corrected prediction matches the desired label.

The optimization problem is formulated as:

$$\min_{\Delta\theta} \|\Delta\theta\|^2 \quad \text{subject to} \quad f(x;\, \theta + \Delta\theta) = y_{\text{target}} \tag{1}$$

where $f(x;\theta)$ is the model prediction for input x with parameters θ, y_{target} is the desired output, and $\Delta\theta$ is the parameter adjustment. The optimization is solved using projected gradient descent with early stopping, implemented via PyTorch's autograd engine [7].

This capability enables users to isolate the effect of specific model components by comparing the magnitude of adjustments required when modifying encoder, attention, or classifier parameters separately. Analysis of individual attention heads [18] confirms that some heads bear disproportionate responsibility for specific predictions, and this observation guides the comparative analysis workflow in our tool.

3.6 Syntactic Dependency Overlay

To investigate the relationship between linguistic structure and learned attention patterns, the framework overlays syntactic dependency structures onto attention visualizations. Using Stanford CoreNLP [20], we parse input sentences and project dependency arcs onto the token sequences. These arcs are then superimposed on attention graphs, with color coding indicating dependency type (subject, object, modifier, etc.). Probing work [13] has demonstrated that syntactic information is encoded in transformer hidden states, which this overlay makes directly inspectable.

This overlay enables grammar-guided simplification, allowing users to mask or highlight attention connections that correspond to specific syntactic relationships. Users can test hypotheses about whether the model attends to linguistically meaningful alignments or spurious correlations.

4. Implementation

The framework is implemented as a lightweight Python library that integrates with PyTorch [7] models through forward and backward hooks. The visualization server uses Flask to serve interactive web-based views built with D3.js [19] and React. D3.js was selected for its representation-transparent approach to web visualization, which enables the fine-grained DOM manipulation required by our linked attention views. Communication between the Python backend and JavaScript frontend is handled through WebSocket connections for real-time updates during perturbation experiments.

The optimization module uses PyTorch's autograd to compute gradients and apply parameter updates, following the constrained optimization formulation inspired by MIRA [6]. To ensure responsiveness during interactive exploration, we cache intermediate representations and use incremental update strategies when possible.

5. Evaluation

We evaluated the framework with six NLP researchers familiar with transformer-based models [1, 17] for natural language inference. Participants completed a series of analytical tasks while thinking aloud, and we collected qualitative feedback on usability and utility, following evaluation protocols common in visual analytics systems [4].

5.1 Task Performance

Participants successfully completed all tasks, including:

- Identifying which attention heads are most responsible for specific predictions
- Diagnosing the cause of prediction errors through attention editing
- Comparing the sensitivity of encoder, attention, and classifier components
- Testing hypotheses about syntactic influence on attention patterns
- Correcting erroneous predictions through minimal parameter adjustment

5.2 Qualitative Feedback

Participants reported that the coordinated multiple views enabled rapid hypothesis formation and testing. The ability to directly manipulate attention weights and observe causal effects was particularly valued, as it provided insight beyond what static attention visualizations offer [14].

Several participants noted that the syntactic dependency overlay revealed unexpected relationships between grammar and attention, consistent with findings that substantial syntactic information is encoded in transformer representations [13]. This suggests directions for future research on linguistically informed model analysis.

Table 1: User evaluation ratings (1–5 scale, mean ± std)

Aspect	Rating
Ease of use	4.3 ± 0.5
Visualization clarity	4.5 ± 0.4
Perturbation responsiveness	4.2 ± 0.6
Hypothesis testing support	4.6 ± 0.3
Integration with existing models	4.1 ± 0.5
Overall utility	4.7 ± 0.2

6. Discussion

The perturbation-driven visual analytics framework addresses a critical gap in tools for interpreting neural language models. By enabling direct manipulation of inputs, internal representations, and outputs, the system supports causal reasoning about model behavior rather than passive observation of static visualizations [14, 15].

The constrained optimization procedure for error correction provides a novel approach to sensitivity analysis, enabling users to quantify the relative importance of different model components for specific predictions. This connects to the finding that specialized attention heads carry disproportionate functional load [18]: our tool allows users to verify such claims interactively for their own models and datasets. This capability has potential applications in model debugging, knowledge distillation, and fine-tuning.

The syntactic dependency overlay represents an initial step toward integrating linguistic theory with model analysis. Probing classifiers have established that syntactic structure is recoverable from transformer hidden states [13]; our framework makes this recoverable structure visually explorable. Future work could extend this to incorporate semantic role labeling, coreference resolution, and discourse structure.

6.1 Limitations

The current implementation supports only transformer-based models for natural language inference, though the architecture generalizes to other architectures and tasks. The optimization procedure, while effective, can be slow for large models due to the need for multiple forward and backward passes.

7. Conclusion and Future Work

This paper introduced a perturbation-driven visual analytics framework for probing internal representations in neural language models [17]. By integrating coordinated multiple views with interactive manipulation capabilities, the system supports hypothesis formation and causal reasoning about model behavior. The

constrained optimization procedure, grounded in the MIRA framework [6], enables minimal-perturbation error correction and comparative analysis of model components.

Future work will extend the framework to support additional model architectures and tasks, incorporate more sophisticated linguistic annotations, and develop collaborative features for team-based model analysis. We also plan to investigate automatic perturbation strategies that suggest informative interventions based on model uncertainty and attention patterns, drawing inspiration from the global attention analysis of AttentionViz [10].

The framework is available as an open-source Python library, lowering the barrier to adoption for routine model interrogation in research and development settings.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

[2] Sarthak Jain and Byron C. Wallace. Attention is not explanation. 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 3543–3556, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357/.

[3] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL https://aclanthology.org/D19-1002/.

[4] Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 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, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.15. URL https://aclanthology.org/2020.emnlp-demos.15/.

[5] Jesse 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*, pages 37–42, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3007. URL https://aclanthology.org/P19-3007/.

[6] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003. URL https://jmlr.csail.mit.edu/papers/v3/crammer03a.html.

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[8] Cheonbok Park, Inyoup Na, Yongjang Jo, Sungbok Shin, Jaehyo Yoo, Bum Chul Kwon, Jian Zhao, Hyungjong Noh, Yeonsoo Lee, and Jaegul Choo. SANVis: Visual analytics for understanding self-attention networks. In *2019 IEEE Visualization Conference (VIS)*, pages 146–150. IEEE, 2019. doi: 10.1109/VISUAL.2019.8933677. URL https://ieeexplore.ieee.org/document/8933677/.

[9] Joseph F. DeRose, Jiayao Wang, and Matthew Berger. Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1160–1170, 2020. doi: 10.1109/TVCG.2020.3028976. URL https://ieeexplore.ieee.org/document/9224153/.

[10] Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. AttentionViz: A global view of transformer attention. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):262–272, 2024. doi: 10.1109/TVCG.2023.3327163. URL https://arxiv.org/abs/2305.03210.

[11] Valentina Neu, Barbara Plank, Thomas Kosch, and Daniel Buschek. Exploring the neural landscape: Visual analytics of neuron activation in large language models with NeuronautLLM. *Information Visualization*, 2024. doi: 10.1177/14738716241283102. URL https://www.sciencedirect.com/science/article/pii/S1524070324000262.

[12] Raymond Li, Wen Xiao, Lanjun Wang, Hyeju Jang, and Giuseppe Carenini. T3-Vis: Visual analytic for training and fine-tuning transformers in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 220–230, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-demo.26. URL https://aclanthology.org/2021.emnlp-demo.26/.

[13] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422. URL https://direct.mit.edu/coli/article/48/1/207/107571/.

[14] Beatrice La Rosa, Graziano Blasilli, Romain Bourqui, David Auber, Giuseppe Santucci, Roberto Capobianco, Enrico Bertini, Romain Giot, and Marco Angelini. State of the art of visual analytics for eXplainable deep learning. *Computer Graphics Forum*, 42(1):319–355, 2023. doi: 10.1111/cgf.14733. URL https://onlinelibrary.wiley.com/doi/full/10.1111/cgf.14733.

[15] Gulsum Alicioglu and Bo Sun. A survey of visual analytics for explainable artificial intelligence methods. *Computers & Graphics*, 102:502–520, 2022. doi: 10.1016/j.cag.2021.09.002. URL https://dl.acm.org/doi/10.1016/j.cag.2021.09.002.

[16] Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. Gradient-based analysis of NLP models is manipulable. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 247–258, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.24. URL https://aclanthology.org/2020.findings-emnlp.24/.

[17] 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*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[18] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580/.

[19] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185. URL https://dl.acm.org/doi/10.1109/TVCG.2011.185.

[20] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-5010. URL https://aclanthology.org/P14-5010/.