Fine-Grained Prototype-Based Interpretability for Operational Text Classification

Bowen Wei
George Mason University
bwei2@gmu.edu

Jinhao Pan
George Mason University
jpan23@gmu.edu

Ziwei Zhu
George Mason University
zzhu20@gmu.edu

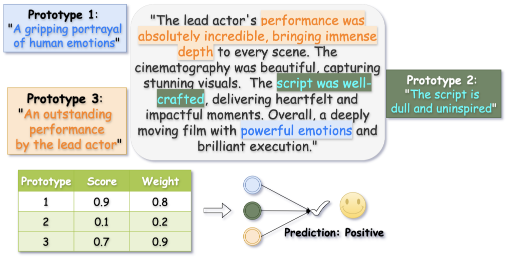
Abstract

We study interpretable, decision-centric NLP for operational settings that require accountability and robustness under uncertainty. We introduce *ProtoLens*, a prototype-based model that produces fine-grained (sub-sentence) rationales aligned to semantically coherent prototypes, enabling principled integration with OR-style decision rules (e.g., cost- and risk-sensitive thresholds, audits, and overrides). Across text classification benchmarks, ProtoLens provides interpretable, human-aligned explanations while matching or exceeding competitive baselines.

1 Introduction

Deep neural networks (DNNs) have achieved remarkable success in text classification tasks [18, 21, 1], but their black-box nature limits deployment in operational contexts requiring interpretability for uncertainty quantification and risk assessment [5, 33]. In domains like customer service operations and financial risk analysis, understanding *why* models make specific predictions is crucial for robust, uncertainty-aware decision-making. While post-hoc methods lack consistency for reliable deployment [14, 24], inherently interpretable models provide the transparency needed for principled decisions under uncertainty [25].

Prototype-based methods bridge ML capabilities and OR principles by generating predictions through comparison with prototypical examples, providing structured frameworks aligned with OR's emphasis on interpretable decision rules. Though extensively studied in computer vision [8, 22, 28, 43], their application to NLP for operational decision-making remains limited [12, 23, 38]. These models offer intuitive interpretability crucial for risk-aware decisions – for instance, using prototypical customer reviews to classify sentiment and inform resource allocation.


Prototype 1:
"A gripping portrayal of human emotions"
Prototype 2:
"The script is dull and uninspired"
Prototype 3:
"An outstanding performance by the lead actor"

"The lead actor's performance was absolutely incredible, bringing immense depth to every scene. The cinematography was beautiful, capturing stunning visuals. The script was well-crafted, delivering heartfelt and impactful moments. Overall, a deeply moving film with powerful emotions and brilliant execution."

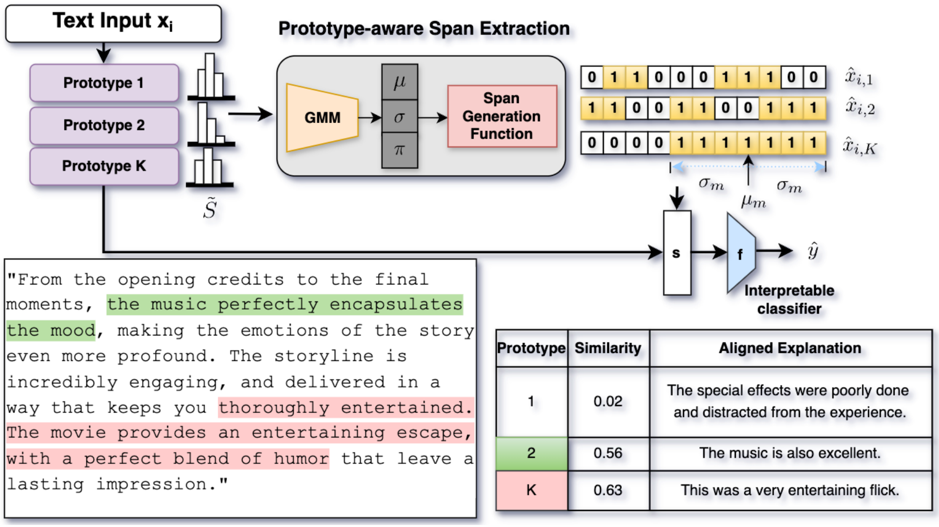
Prototype	Score	Weight
1	0.9	0.8
2	0.1	0.2
3	0.7	0.9

Figure 1: Interpretable Classification.

However, existing prototype-based models typically operate at the instance/sentence level, lacking the granularity needed for actionable insights in complex operational scenarios [12, 23]. Consider customer feedback like "The visuals were stunning, but the plot was too predictable" – instance-level prototypes miss nuances critical for targeted interventions (e.g., maintaining visual quality while improving plot development). This limitation prevents granular decision-making and feature-level uncertainty quantification essential for operational responses.

We propose ProtoLens, a novel prototype-based model integrating ML flexibility with OR's structured decision frameworks for fine-grained, uncertainty-aware operational support. As illustrated in

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.



"From the opening credits to the final moments, the music perfectly encapsulates the mood, making the emotions of the story even more profound. The storyline is incredibly engaging, and delivered in a way that keeps you thoroughly entertained. The movie provides an entertaining escape, with a perfect blend of humor that leave a lasting impression."

Prototype	Similarity	Aligned Explanation
1	0.02	The special effects were poorly done and distracted from the experience.
2	0.56	The music is also excellent.
K	0.63	This was a very entertaining flick.

Figure 2: Model Structure.

Figure 1, ProtoLens extracts prototype-specific sub-sentence spans, enabling transparent decision support through selective prototype activation.

ProtoLens embeds OR insights through two core modules: (1) **Prototype-aware Span Extraction** using Dirichlet Process Gaussian Mixture Models (DPGMM) [11, 30] for probabilistic span extraction with uncertainty quantification, and (2) **Prototype Alignment** ensuring prototypes remain semantically and operationally meaningful throughout training.

Extensive experiments demonstrate ProtoLens outperforms baselines while providing actionable, uncertainty-aware explanations suitable for operational contexts where interpretability and risk assessment are paramount.

2 Method

To deliver inherently interpretable predictions at a fine-grained level, we introduce **ProtoLens**, a prototype-based interpretable neural network. ProtoLens is designed to overcome two primary challenges: **(C1) How to effectively extract text spans associated with a given prototype to provide interpretable predictions?** and **(C2) How to ensure learned prototypes are semantically reasonable and effective for interpretation?** To address C1, we propose a Prototype-Aware Span Extraction module, which extracts most relevant text spans for prototypes by a Dirichlet Process Gaussian Mixture Model. To address C2, we design a Prototype Alignment mechanism to adaptively align prototype embeddings to representative data samples through training. The overall model architecture is illustrated in Figure 2. A detailed analysis of prototype alignment and the learning objectives is provided in Appx. C and Appx. D.

2.1 Overall Structure

Given a corpus of textual data $\mathcal{D} = \{(x_i, y_i)\}$, where $i = 1, \ldots, N$, each instance x_i is associated with a label $y_i \in \mathcal{Y}$, our model processes the text through a text encoder, such as BERT [7], $\psi : \mathcal{X} \to \mathbb{R}^d$, where $\mathcal{X}$ represents the space of inputs and d is determined by the encoder.

For a text instance x, it is first inputted to an **Prototype-aware Span Extraction** module, containing a set of trainable prototypes $\mathcal{P} = \{\mathbf{p}_k \in \mathbb{R}^d : k = 1, \ldots, K\}$, where each prototype is represented by a learnable embedding, and the hyperparameter K is the number of prototypes specified. The model will deliver classifications by comparing the input to these prototypes. For each prototype k, we identify a relevant text span $x^k \subseteq x$, which represents a sub-sentence capturing the most relevant portion of x associated with that prototype. We then use an encoder ψ to compute an embedding for each extracted span x^k: $\mathbf{z}^k = \psi(x^k)$.

The similarity between $\mathbf{z}^k$ and prototype $\mathbf{p}_k$ is then computed as $s^k = \text{RMSNorm}(cos(\mathbf{z}^k, \mathbf{p}_k))$. The final prediction is computed via an interpretable model f applied to the similarity vector across all prototypes $\mathbf{s} = [s^1, s^2, \ldots, s^K]$: $\hat{y} = f(\mathbf{s})$, where $\mathbf{s}$ captures the proximity to all prototypes, serving as features for the final prediction; and f can be any interpretable models, such as decision tree or logistic regression. In this paper, we adopt the logistic regression as f.

Model Interpretation. The interpretability of ProtoLens is two-fold. First, it employs prototypes aligned with real-world text sentences to represent human-understandable concepts, assigning weights that reveal their presence and importance in predictions, ensuring intrinsic interpretability. Second, it extracts input spans most relevant to the activated prototypes, allowing users to intuitively compare these spans with the corresponding prototypes for fine-grained interpretability. These prototypes serve as features for an interpretable classifier, such as logistic regression, which provides an additional layer of transparency. Logistic regression assigns interpretable coefficients to each prototype, offering clear insights into how each prototype contributes to the final prediction. As illustrated in Figure 1, ProtoLens highlights spans from the input text that relevant to prototypes. For example, spans like "powerful emotions" and "script was well-crafted" align with Prototype 1 and Prototype 3, respectively, contributing positively to the prediction. In contrast, Prototype 2, "The script is dull and uninspired", is not activated and thus has no contribution to the prediction.

2.2 Prototype-aware Span Extraction

To extract the most relevant spans of the input text x for each prototype, we divide the input x into n-grams $x = (c_t)_{t=1}^{T}$, where c_t denotes the t-th n-gram, T is the total number of n-grams, and n is a hyperparameter. A text span is composed of consecutive n-grams. The text encoder processes each part $c_t \in x$ to produce an embedding $\mathbf{e}_t = \psi(c_t) \in \mathbb{R}^d$. The similarity $m_{t,k}$ between the part embedding $\mathbf{e}_t$ and the prototype embedding $\mathbf{p}_k$ is then measured using cosine similarity: $m_{t,k} = cos(\mathbf{e}_t, \mathbf{p}_k)$. The intermediate output of the module is the similarity vector between each text input and prototype k, denoted as $\mathbf{m}_k = (m_{t,k})_{t=1}^{T}$.

2.2.1 Similarity Distribution Modeling by DPGMM

Identifying the most relevant text spans that align with a prototype is a challenging task due to the inherent complexity and variability of patterns in natural language. The primary aim of employing "fine-grained prototypes" is to extract text spans of flexible lengths, rather than relying on rigid instance/sentence-level, or fixed-size windows.

To address this challenge, we use a Dirichlet Process Gaussian Mixture Model (DPGMM) [11, 30], which represents the relevance between prototypes and text spans as a probability distribution. By modeling similarity distributions in $\mathbf{m}_k$ with Gaussian components, DPGMM provides an effective framework for dynamically identifying high-similarity regions in the input text, thereby facilitating the extraction of flexible and relevant text spans. DPGMM approximates $\mathbf{m}_k$ using up to G Gaussian components: $p(\mathbf{m}_k) = \sum_{g=1}^{G} \pi_g \cdot \mathcal{N}(\mathbf{m}_k \mid \mu_g, \sigma_g)$, where π_g is the mixture weight, and $\mathcal{N}(\mathbf{m}_k \mid \mu_g, \sigma_g)$ is the Gaussian distribution with mean μ_g and standard deviation σ_g.

We deploy a neural network based method to learn these parameters following existing works [42, 4]. Specifically, we first learn a hidden representation $\mathbf{h} = \text{MLP}(\mathbf{m}_k)$ and compute these parameters as: $\boldsymbol{\mu} = \text{sigmoid}(\mathbf{W}_\mu \mathbf{h} + \mathbf{b}_\mu) \times T$, $\boldsymbol{\sigma} = \exp(\mathbf{W}_\sigma \mathbf{h} + \mathbf{b}_\sigma)$, $\boldsymbol{\nu} = \text{sigmoid}(\mathbf{W}_\pi \mathbf{h} + \mathbf{b}_\pi)$, and $\pi_g = \nu_g \prod_{\ell=1}^{g-1}(1 - \nu_\ell)$ for $g = 1, \ldots, G$. Here, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the parameters for the Gaussian components, while $\boldsymbol{\pi}$ is determined using the Stick-Breaking Process [31], allowing for an adaptive number of components. Detailed explanations can be found in Appendix B.

2.2.2 Span Extraction

To extract a span that focuses on the most relevant area of the text, we select the Gaussian component with the highest mixture weight $\pi_g = \max(\pi)$, characterized by (μ_g, σ_g). Then, μ_g serves as the center of the span, while σ_g defines its length. The span is thus given by: $x^k = x[\mu_g - \sigma_g, \mu_g + \sigma_g]$.

3 Experiments

To deliver inherently interpretable, fine-grained predictions, we present **ProtoLens**, a prototype-based network addressing two challenges: **(C1)** extracting prototype-relevant text spans and **(C2)** ensuring semantically coherent prototypes. For **C1**, *Prototype-Aware Span Extraction* uses a Dirichlet Process

Table 1: Performance of ProtoLens in comparison with baselines.

Model	IMDB	Amazon	Yelp	Hotel	Steam	DBPedia	Consumer
Llama-3-8b	0.813	0.767	0.787	0.787	0.667	0.768	0.807
MPNet	0.846	0.899	0.950	0.961	0.913	0.991	0.933
Bag-of-words	0.877	0.830	0.908	0.905	0.844	0.993	0.930
ProSeNet	0.863	0.875	0.932	0.930	0.834	0.984	0.878
ProtoryNet	0.871	0.890	0.941	0.949	0.876	0.991	0.927
ProtoLens(MPNet)	**0.903**	**0.937**	**0.962**	**0.963**	**0.931**	**0.995**	**0.945**

Gaussian Mixture Model (DPGMM) to select salient sub-sentence spans. For **C2**, *Prototype Alignment* adaptively anchors prototype embeddings to representative samples during training. Additional analyses, including hyperparameter sensitivity and ablation studies, are provided in Appx. J I.

3.1 Prediction Accuracy

We evaluate the accuracy of ProtoLens against several competitive baselines, including both prototype-based and non-prototype-based methods. The results are presented in Table 1. ProtoLens consistently achieves the highest scores, outperforming the baselines in all cases. The consistently higher performance of ProtoLens demonstrates its effectiveness and robustness across diverse domains, highlighting its superiority in leveraging fine-grained interpretability without sacrificing accuracy.

3.2 Model Interpretations

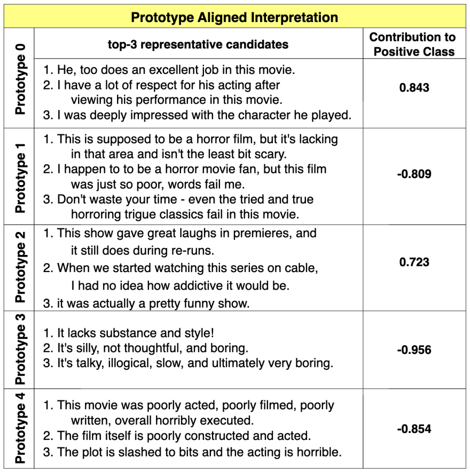
Two-fold interpretability. ProtoLens provides (i) *prototype-level* interpretations by aligning learned prototypes to representative training sentences with weights, exposing which concepts are present and how strongly they contribute; and (ii) *span-level* interpretations by extracting input spans most relevant to activated prototypes, enabling fine-grained, example-based rationales.

3.2.1 Prototype Interpretation

Trained on IMDB with $K{=}10$ prototypes, ProtoLens learns concise, human-readable concepts (e.g., acting, horror, humor, storyline, execution). Fig. 3 shows five randomly selected prototypes and their aligned sentences. Despite their brevity, these prototypes capture salient factors and enable quick inspection of the model's reasoning. Additional examples appear in Appx. G.

3.2.2 Classification Interpretation

For a test input, ProtoLens extracts the most relevant span per prototype, computes span–prototype similarity, and aggregates weighted activations into a prediction. In a positive IMDB example (Fig. 9), the top activations emphasize entertainment (*"highly entertaining flick"*, sim. 0.708, weight 0.985), humor (*"crime comedy … very funny"*, 0.549, 0.247), and acting (*"some great actors … "*, 0.730, 0.931), yielding POSITIVE. In a negative example (Fig. 11), activations highlight cheap effects (*"cheap special effects"*, 0.657, −0.956), frustration (*"watching it the whole 2 hours"*, 0.676, −0.809), and missing character development (*"no character development"*, 0.664, −0.756), yielding NEGATIVE. More cases are in Appx. G.



Prototype Aligned Interpretation		
	top-3 representative candidates	Contribution to Positive Class
Prototype 0	1. He, too does an excellent job in this movie. 2. I have a lot of respect for his acting after viewing his performance in this movie. 3. I was deeply impressed with the character he played.	0.843
Prototype 1	1. This is supposed to be a horror film, but it's lacking in that area and isn't the least bit scary. 2. I happen to to be a horror movie fan, but this film was just so poor, words fail me. 3. Don't waste your time - even the tried and true horroring trigue classics fail in this movie.	-0.809
Prototype 2	1. This show gave great laughs in premieres, and it still does during re-runs. 2. When we started watching this series on cable, I had no idea how addictive it would be. 3. it was actually a pretty funny show.	0.723
Prototype 3	1. It lacks substance and style! 2. It's silly, not thoughtful, and boring. 3. It's talky, illogical, slow, and ultimately very boring.	-0.956
Prototype 4	1. This movie was poorly acted, poorly filmed, poorly written, overall horribly executed. 2. The film itself is poorly constructed and acted. 3. The plot is slashed to bits and the acting is horrible.	-0.854

Figure 3: Aligned prototypes with training sentences.

4 Conclusion

In this paper, we present ProtoLens, a prototype-based model offering fine-grained, sub-sentence level interpretability for text classification. we introduce a Prototype-aware Span Extraction module with a Prototype Alignment mechanism to ensure prototypes remain semantically meaningful and aligned with human-understandable examples. Extensive experiments across multiple benchmarks show that ProtoLens outperforms both prototype-based and non-interpretable baselines in accuracy while providing intuitive and detailed explanations.

References

[1] Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3), 2012.

[2] Sercan O Arik and Tomas Pfister. Protoattend: Attention-based prototypical learning. *Journal of Machine Learning Research*, 21(210):1–35, 2020.

[3] Dzmitry Bahdanau. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Christopher M Bishop. Mixture density networks. 1994.

[5] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[6] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[8] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, page 4, 2018.

[9] Richard Fikes and Tom Kehler. The role of frame-based representation in reasoning. *Communications of the ACM*, 28(9):904–920, 1985.

[10] Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba Salahuddin, Robert Jenssen, Marina Höhne, and Michael Kampffmeyer. Protovae: A trustworthy self-explainable prototypical variational model. *Advances in Neural Information Processing Systems*, 35:17940–17952, 2022.

[11] Dilan Görür and Carl Edward Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664, 2010.

[12] Dat Hong, Tong Wang, and Stephen Baek. Protorynet-interpretable text classification via prototype trajectories. *Journal of Machine Learning Researcyh*, 24(264):1–39, 2023.

[13] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.

[14] Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. Understanding convolutional neural networks for text classification. *arXiv preprint arXiv:1809.08037*, 2018.

[15] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

[16] Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. *arXiv preprint arXiv:1911.06194*, 2019.

[17] Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. *Advances in neural information processing systems*, 27, 2014.

[18] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[19] Junbing Li, Changqing Zhang, Joey Tianyi Zhou, Huazhu Fu, Shuyin Xia, and Qinghua Hu. Deep-lift: Deep label-specific feature learning for image annotation. *IEEE transactions on Cybernetics*, 52(8):7732–7741, 2021.

[20] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[21] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.

[22] Yao Ming, Panpan Xu, Furui Cheng, Huamin Qu, and Liu Ren. Protosteer: Steering deep sequence model with prototypes. *IEEE transactions on visualization and computer graphics*, 26(1):238–248, 2019.

[23] Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 903–913, 2019.

[24] Saumitra Mishra, Bob L Sturm, and Simon Dixon. Local interpretable model-agnostic explanations for music content analysis. In *ISMIR*, volume 53, pages 537–543, 2017.

[25] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[26] W James Murdoch, Peter J Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. *arXiv preprint arXiv:1801.05453*, 2018.

[27] Mike A Nalls, Jose Bras, Dena G Hernandez, Margaux F Keller, Elisa Majounie, Alan E Renton, Mohamad Saad, Iris Jansen, Rita Guerreiro, Steven Lubbe, et al. Neurox, a fast and efficient genotyping platform for investigation of neurodegenerative diseases. *Neurobiology of aging*, 36(3):1605–e7, 2015.

[28] Meike Nauta, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2744–2753, 2023.

[29] Zhongang Qi, Saeed Khorram, and Fuxin Li. Visualizing deep networks by optimizing with integrated gradients. In *CVPR workshops*, volume 2, pages 1–4, 2019.

[30] Carl Rasmussen. The infinite gaussian mixture model. *Advances in neural information processing systems*, 12, 1999.

[31] Lu Ren, Lan Du, Lawrence Carin, and David B Dunson. Logistic stick-breaking process. *Journal of Machine Learning Research*, 12(1), 2011.

[32] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.

[33] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

[34] Rory Sayres, Ankur Taly, Ehsan Rahimy, Katy Blumer, David Coz, Naama Hammel, Jonathan Krause, Arunachalam Narayanaswamy, Zahra Rastegar, Derek Wu, et al. Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy. *Ophthalmology*, 126(4):552–564, 2019.

[35] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020.

[36] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding, 2020.

[37] Pål Sørgaard. Evaluating expert system prototypes. *AI & society*, 5:3–17, 1991.

[38] Zhivar Sourati, Darshan Deshpande, Filip Ilievski, Kiril Gashteovski, and Sascha Saralajew. Robust text classification: Analyzing prototype-based networks. *arXiv preprint arXiv:2311.06647*, 2023.

[39] Yee Whye Teh et al. Dirichlet process. *Encyclopedia of machine learning*, 1063:280–287, 2010.

[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[41] Michael Tsang, Youbang Sun, Dongxu Ren, and Yan Liu. Can i trust you more? model-agnostic hierarchical explanations. *arXiv preprint arXiv:1812.04801*, 2018.

[42] Cinzia Viroli and Geoffrey J McLachlan. Deep gaussian mixture models. *Statistics and Computing*, 29:43–51, 2019.

[43] Mengqi Xue, Qihan Huang, Haofei Zhang, Lechao Cheng, Jie Song, Minghui Wu, and Mingli Song. Protopformer: Concentrating on prototypical parts in vision transformers for interpretable image recognition. *arXiv preprint arXiv:2208.10431*, 2022.

[44] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1:43–52, 2010.

A Related Work

Post-hoc Explanations. Several post-hoc methods interpret DNN models by analyzing gradients or neuron activations, such as Integrated Gradients [34, 29], DeepLift [19], and NeuroX [27]. **(author?)** [41] proposed a hierarchical method to capture interaction effects, later adapted by **(author?)** [16] for text classification. In sentiment analysis, contextual decomposition [26] identifies sentiment words and their contributions. Attention-based models, such as **(author?)** [3, 32], analyze attention weights, though **(author?)** [15] question their explanatory power.

Prototype-based Deep Neural Networks. Prototype-based deep neural networks enhance interpretability by using prototypes as intuitive references for decision-making, a concept rooted in traditional models [37, 9, 17]. While prototype-based reasoning has been extensively developed in CV, with methods like ProtoPNet [6] for image classification and ProtoVAE [10] introducing diverse and interpretable prototypes, its application in NLP is a relatively new area. Early works such as ProSeNet [23] adapted prototype-based reasoning for text classification, followed by ProtoAttend [2], which employed attention mechanisms for dynamic prototype selection. Recently, ProtoryNet [12] introduced prototype trajectory modeling to improve interpretability across domains. Despite these advances, prototype-based approaches in NLP remain underexplored, making our work a significant step forward in this emerging field.

Unlike previous methods, our approach directly embeds interpretability at the sub-sentence level, providing more granular insights than word- or sentence-level methods.

B DPGMM

To model each similarity distribution as a mixture of Gaussian components, we use a neural network that takes a hidden representation h as input, which is derived from $\mathbf{m}_k$ via a two-layer MLP: $h = \text{MLP}(\mathbf{m}_k)$. This hidden representation h is then used to generate the parameters of the Gaussian mixture, including the mixture weights $\boldsymbol{\pi}$, means $\boldsymbol{\mu}$, and standard deviations $\boldsymbol{\sigma}$, allowing the model to approximate the similarity distribution effectively.

Means (μ) and **Standard Deviations** (σ)**.** The parameters of the Gaussian components are computed as follows:

$$\mu = \text{sigmoid}(\mathbf{W}_\mu h + \mathbf{b}_\mu) \times T, \tag{1}$$

$$\sigma = \exp(\mathbf{W}_\sigma h + \mathbf{b}_\sigma), \tag{2}$$

where μ and σ are the mean and standard deviation for each of the M Gaussian components.

Mixture Weights (π). To dynamically determine the mixture weights, we employ the Stick-Breaking Process [31], with the Dirichlet Process (DP) [39] implicitly implemented through the stick-breaking formulation. The DP provides a nonparametric Bayesian approach that allows the model to determine the appropriate number of components adaptively, which is crucial for handling data with unknown complexity.

We define a maximum number of Gaussian components, G, which represents the potential number of components for approximating the similarity distribution. The mixture weights π_g for each component g are generated as follows:

$$\nu_g = \text{sigmoid}(\mathbf{W}_\pi h + \mathbf{b}_\pi), \tag{3}$$

$$\pi_g = \nu_g \prod_{\ell=1}^{g-1} (1 - \nu_\ell), \quad g = 1, \ldots, G, \tag{4}$$

Here, ν_g is computed by applying a sigmoid function to a linear transformation of the hidden representation h. The Stick-Breaking Process ensures that the mixture weights π_m sum to one and adaptively determine the number of active components, enabling the model to capture complex and potentially multi-modal distributions.

C Prototype Alignment

To ensure interpretable classifications, the learned prototypes must be semantically meaningful. However, these prototypes are numerical embeddings that are not inherently interpretable by human users. Therefore, we introduce a prototype alignment mechanism that maps each prototype to real-world training text sentences throughout the learning process.

Representative Candidates. We begin by encoding all sentences in the training instances (an instance can contain multiple sentences) into embeddings. In the embedding space, we apply the k-means to cluster sentences. The top 50 sentences closest to each cluster center obtained from k-means serve as representative examples of each cluster, making them suitable candidates for aligning prototypes.

At one epoch during training, for each prototype with its current learned embedding $\mathbf{p}_k$, the top 3 most similar candidate sentences (green circles) from the representative candidates are selected. These candidates are averaged to form a representative embedding $\mathbf{c}_k$ (purple cross), which encapsulates the meaning from actual training data. The prototype is then updated towards $\mathbf{c}_k$ (orange arrow), resulting in an updated prototype $\mathbf{p}'_k$ (yellow star).

Specifically, $\mathbf{p}_k$ is updated towards $\mathbf{c}_k$ controlled by a weight factor $\omega_k = \text{sigmoid}(\gamma \cdot (d_k - \tau))$, where d_k represents the Euclidean distance between $\mathbf{p}_k$ and $\mathbf{c}_k$, τ is the movement threshold and γ controls the smoothness of the transition. The updated prototype $\mathbf{p}'_k$ is derived as a weighted combination: $\mathbf{p}'_k = \omega_k \cdot (\mathbf{p}_k + \tau \cdot \mathbf{u}_k) + (1 - \omega_k) \cdot \mathbf{c}_k$, where $\mathbf{u}_k = \frac{\mathbf{c}_k - \mathbf{p}_k}{d_k + \epsilon}$ is the unit vector pointing from $\mathbf{p}_k$ to $\mathbf{c}_k$, with ϵ being a small value to prevent division by zero. If $\mathbf{p}_k$ is far from $\mathbf{c}_k$ (i.e., $d_k \geq \tau$), $\mathbf{p}_k$ will move a distance of τ toward $\mathbf{c}_k$. Conversely, if $d_k \leq \tau$, $\mathbf{p}_k$ is directly aligned with $\mathbf{c}_k$. This process ensures that the prototypes shift toward semantically meaningful regions without abrupt changes.

D Learning Objectives

The learning objectives of the proposed model consist of three key components that contribute to both prediction accuracy and the interpretability of the learned representations.

D.1 GMM Loss

To approximate complex similarity distributions between text samples and prototypes, we employ a Negative Log-Likelihood (NLL) loss for GMM jointly trained with the model: $\mathcal{L}_{\text{NLL}} = -\log(\sum_{m=1}^{M} \pi_m \cdot \mathcal{N}(\tilde{s} \mid \mu_m, \sigma_m) + \epsilon)$, where π_m, μ_m, and σ_m are the mixture weights, means, and standard deviations of the m-th Gaussian component, respectively, and ϵ is a small constant added for numerical stability.

The overall loss for the GMM is defined as: $\mathcal{L}_{\text{GMM}} = \mathbf{E}[\mathcal{L}_{\text{NLL}}] + \mathcal{L}_{\text{L1}}$, where an L_1 regularization term is introduced to promote sparsity in the mixture weights: $\mathcal{L}_{\text{L1}} = \lambda \sum_{m=1}^{M} |\pi_m|$, where λ controls the regularization strength. This sparsity encourages the model to focus on a few significant Gaussian components. λ is set to $1e^{-3}$ for all experiments.

D.2 Diversity Loss

To encourage the model to learn high-quality and diverse prototypes, we introduce a **Diversity Loss** based on cosine distance: $\mathcal{L}_{\text{div}} = \sum_{i \neq j}(1 - \cos(\mathbf{p}_i, \mathbf{p}_j))$. Maximizing this diversity loss enhances generalization and interpretability by maintaining a diverse set of prototypes.

D.3 Overall Objective

The final objective function for the proposed model is a weighted combination of the aforementioned loss components: $\mathcal{L} = \text{CrossEntropy}(y, \hat{y}) + \alpha\mathcal{L}_{\text{GMM}} - \beta\mathcal{L}_{\text{div}}$, where y represents the true labels, $\hat{y}$ denotes the prediction, α and β are hyperparameters that control the balance between accuracy, Gaussian mixture modeling, and prototype diversity. α and β is set to $1e^{-1}$ and $1e^{-3}$ for all experiments, respectively.

E Datasets

The IMDB dataset contains 25,000 balanced training and test samples and follows a binary sentiment classification format. The dataset was split into training (90%) and validation (10%) partitions. The Yelp Reviews dataset consists of 580,000 samples, with training and test sets comprising 550,000 and 30,000 samples, respectively. Sentiments were binarized by treating 1–2 stars as negative and 3–4 stars as positive. The Amazon dataset was created by selecting 30,000 random reviews, with 24,000 samples allocated for training and validation and 6,000 for testing. The Hotel dataset includes 20,000 reviews evaluating 1,000 hotels, reduced to a balanced subset of 4,508 reviews (2,254 positive and 2,254 negative). The Steam Reviews dataset consists of 130,000 pre-processed reviews, balanced between positive and negative sentiments. Reviews with fewer than 10 characters or containing less than two sentences were excluded.

The DBPedia dataset is a multiclass dataset extracted from Wikipedia. For the experiments in this paper, we use only 4 labels: “Person,” “Animal,” “Building,” and “Natural Place.” Similarly, the Consumer Complaints dataset is a multiclass dataset. For the experiments, we use only 4 classes: “Checking or Savings Account,” “Credit Card or Prepaid Card,” “Debt Collection,” and “Mortgage.”

In all experiments, pre-trained embeddings from the BERT-based language model [35] were employed to convert raw text into sentence embeddings, enabling downstream analysis.

F Experimental Setup

Datasets. We evaluate ProtoLens on seven diverse text classification datasets spanning single-label, multi-label, and domain-specific classification tasks: IMDB, Yelp, Amazon, Hotel, Steam, DBPedia, and Consumer Complaint. Details are provided in Appendix E.

Reproducibility. The ProtoLens model was implemented using PyTorch. For training, the prototype number K is selected from $\{10, 20, 40, 50, 100\}$. The learning rate is selected from $\{1e-4, 1e-5, 5e-5\}$, with a decay of 10% every 10 epochs. We used the AdamW optimizer [20] with a batch size of 16 for 25 epochs and the n-gram size is selected from $\{1, 3, 5, 7, 9\}$. The experiments were conducted on an NVIDIA A100 80GB GPU. Code and data are available at `https://github.com/weibowen555/ProtoLens`.

Baselines. We compare ProtoLens against a range of baselines, encompassing both interpretable and non-interpretable models. The interpretable baselines include ProSeNet [23] and ProtoryNet [12], both are SOTA prototype-based methods that provide insights into their predictions via learned prototypical representations. Additionally, we include a zero-shot Llama-3-8b [40], MPNet [36] and a Bag-of-Words model [44] using TF-IDF representations and Logistic Regression for interpretable classification [13]. The prompt used for LLaMA-3 is provided in Appendix M.

G Prototype Interpretation

To assess the interpretability of the ProtoLens model, we provide prototype-aligned interpretations across multiple datasets. Each figure showcases the top-3 original text sentences from the training set that are most aligned with each prototype. These examples illustrate how ProtoLens associates prototypes with representative samples, making its decision-making process more interpretable and transparent.

For the IMDB dataset, as shown in Figure 4, ProtoLens aligns prototypes with representative training samples that reflect key aspects of movie reviews. Positive prototypes are associated with reviews praising elements such as acting and overall quality, as seen in samples like "He does an excellent job in this movie" and "I deeply enjoyed his performance." Negative prototypes, on the other hand, align with reviews critiquing aspects like plot and execution, exemplified by samples such as "This movie was poorly acted, poorly filmed, poorly written" and "It's talky, illogical, slow, and ultimately boring." These representative samples demonstrate ProtoLens' ability to capture diverse perspectives in sentiment analysis.

In the Yelp dataset, as shown in Figure 5, ProtoLens aligns prototypes with representative samples that capture customer opinions on food, service, and ambiance. Positive prototypes are linked to text such as "The service is impeccable" and "The food is great, good portions and quality," reflecting positive customer experiences. Conversely, negative prototypes correspond to samples highlighting dissatisfaction, such as "The food was horrible" and "The place looked dirty and disorganized." These aligned samples illustrate how ProtoLens effectively represents common patterns in customer feedback.

For the Hotel dataset, as shown in Figure 6, ProtoLens aligns prototypes with representative training samples reflecting both positive and negative experiences. Positive prototypes align with samples such as "Room was clean and good" and "The staff were friendly and helpful," highlighting aspects of comfort and service. Negative prototypes correspond to samples like "The room had no soundproofing" and "The carpet is disgusting and filthy," emphasizing common complaints in hospitality feedback. These representative samples demonstrate ProtoLens' ability to capture recurring themes in hotel reviews.

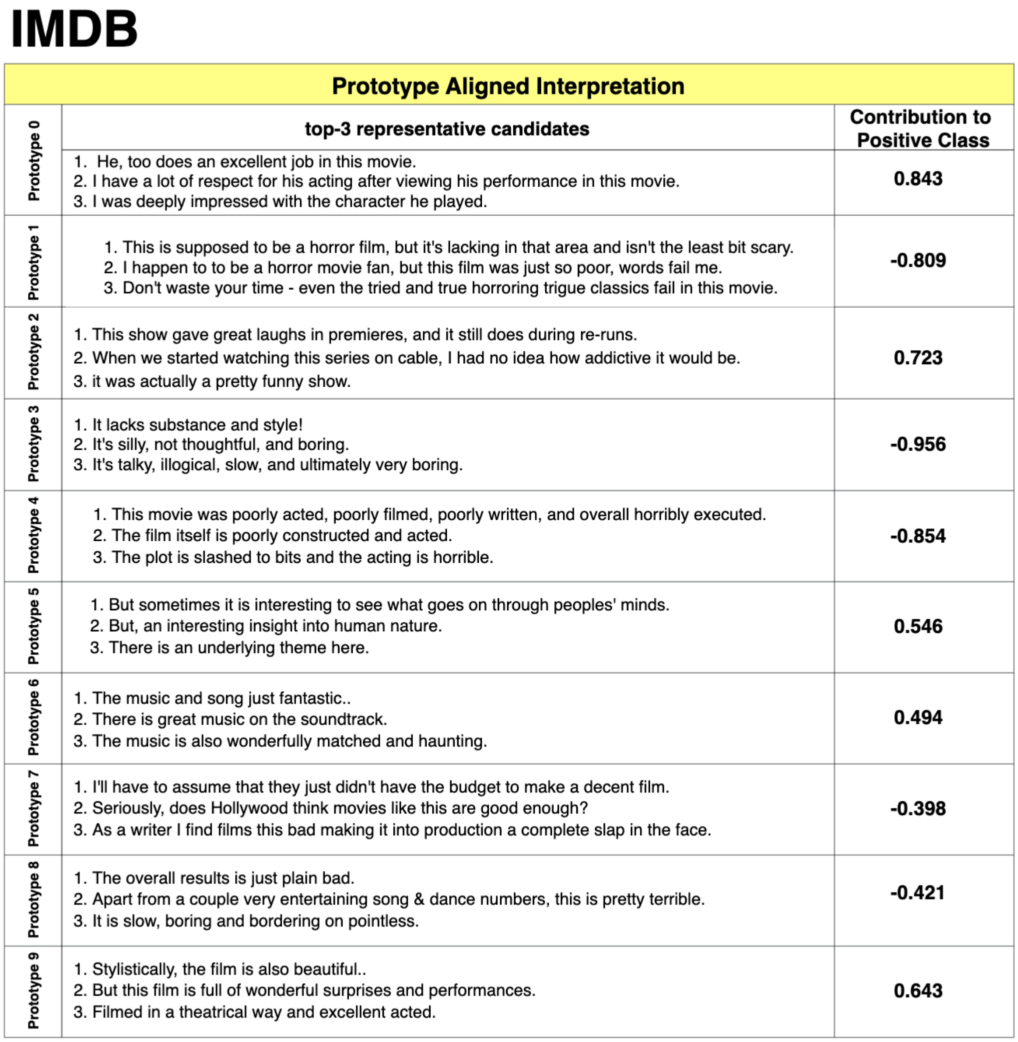
In the Steam dataset, as shown in Figure 7, ProtoLens identifies prototypes aligned with gaming reviews that reflect both satisfaction and dissatisfaction. Positive prototypes are linked to reviews like "This game is amazing" and "Runs smooth even on low settings," which highlight positive gameplay experiences. Negative prototypes, on the other hand, align with samples such as "The servers are abandoned" and "This game sucks, do not buy it," reflecting technical issues and user frustration. These representative samples demonstrate ProtoLens' ability to adapt to highly specific and technical feedback in gaming.

For the Amazon dataset, as shown in Figure 8, ProtoLens aligns prototypes with representative training samples focusing on product quality, usability, and service. Positive prototypes correspond to samples such as "The decor is beautiful and the ambiance is great" and "I enjoyed this place and will go back," reflecting favorable customer feedback. Negative prototypes align with samples like "The food was uninspired and lacked flavor" and "Horrible management and worse customer service," highlighting dissatisfaction. These examples demonstrate ProtoLens' versatility in capturing meaningful patterns in e-commerce reviews.

Overall, these results underscore ProtoLens' ability to align prototypes with semantically meaningful training samples, providing interpretable insights into the patterns learned during training. This interpretability is key to understanding the model's reasoning across diverse datasets.

H Classification Interpretation

ProtoLens explains its classification predictions by aligning input text with prototypes from the training set and computing similarity scores to highlight the most relevant prototypes. Each prototype contributes to the final prediction based on its similarity to the input text and its associated sentiment weight. Below, we discuss how ProtoLens interprets both positive and negative classifications through representative examples.



IMDB

Prototype Aligned Interpretation		
	top-3 representative candidates	Contribution to Positive Class
Prototype 0	1. He, too does an excellent job in this movie. 2. I have a lot of respect for his acting after viewing his performance in this movie. 3. I was deeply impressed with the character he played.	0.843
Prototype 1	1. This is supposed to be a horror film, but it's lacking in that area and isn't the least bit scary. 2. I happen to to be a horror movie fan, but this film was just so poor, words fail me. 3. Don't waste your time - even the tried and true horroring trigue classics fail in this movie.	-0.809
Prototype 2	1. This show gave great laughs in premieres, and it still does during re-runs. 2. When we started watching this series on cable, I had no idea how addictive it would be. 3. it was actually a pretty funny show.	0.723
Prototype 3	1. It lacks substance and style! 2. It's silly, not thoughtful, and boring. 3. It's talky, illogical, slow, and ultimately very boring.	-0.956
Prototype 4	1. This movie was poorly acted, poorly filmed, poorly written, and overall horribly executed. 2. The film itself is poorly constructed and acted. 3. The plot is slashed to bits and the acting is horrible.	-0.854
Prototype 5	1. But sometimes it is interesting to see what goes on through peoples' minds. 2. But, an interesting insight into human nature. 3. There is an underlying theme here.	0.546
Prototype 6	1. The music and song just fantastic.. 2. There is great music on the soundtrack. 3. The music is also wonderfully matched and haunting.	0.494
Prototype 7	1. I'll have to assume that they just didn't have the budget to make a decent film. 2. Seriously, does Hollywood think movies like this are good enough? 3. As a writer I find films this bad making it into production a complete slap in the face.	-0.398
Prototype 8	1. The overall results is just plain bad. 2. Apart from a couple very entertaining song & dance numbers, this is pretty terrible. 3. It is slow, boring and bordering on pointless.	-0.421
Prototype 9	1. Stylistically, the film is also beautiful.. 2. But this film is full of wonderful surprises and performances. 3. Filmed in a theatrical way and excellent acted.	0.643

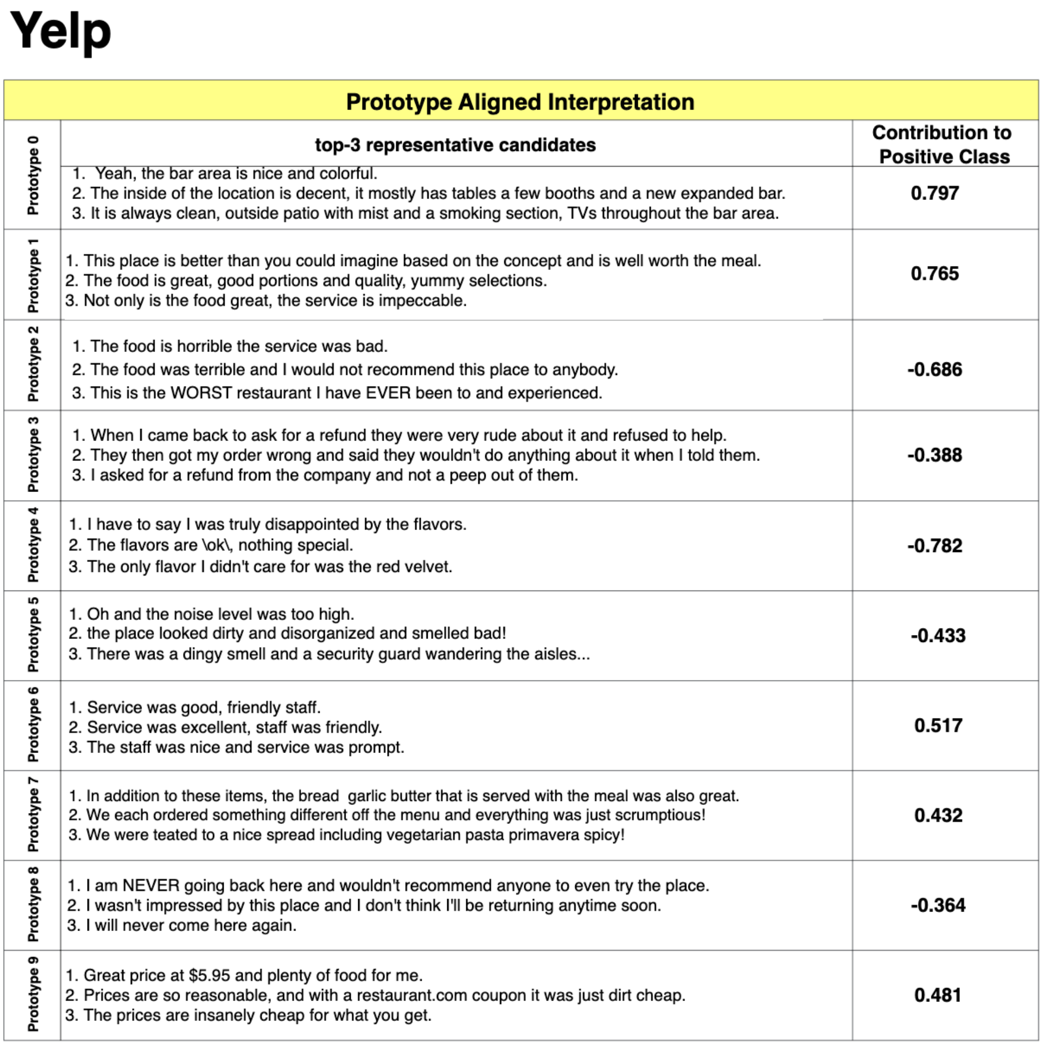
Figure 4: Aligned interpretation of prototypes with corresponding text sentences on the IMDB dataset. Each prototype is associated with specific spans of text and sentiment weights, providing insights into the reasoning behind the model's predictions.

H.1 Positive Sentiment Interpretation

Figure 10 demonstrates a positive sentiment classification. ProtoLens activates three prototypes that correspond to semantically aligned samples from the training set. For instance, **Prototype 10** highlights positive movie reviews with phrases like "In all it is a good movie to see," capturing strong alignment with the input's positive tone. Similarly, **Prototype 14** emphasizes "acting was terrific," contributing further evidence of a positive sentiment. The similarity scores and sentiment weights of these prototypes are combined to determine the final classification as positive. This process underscores how ProtoLens grounds its decisions in interpretable and meaningful text examples.

H.2 Negative Sentiment Interpretation

Figure 12 illustrates a negative sentiment classification. ProtoLens activates prototypes that align with critical text samples from the training set. For example, **Prototype 3** reflects dissatisfaction through statements such as "It's talky, illogical, slow, and ultimately very boring," aligning with the input's description of the movie as "pretty bad." **Prototype 4** further reinforces the negative sentiment by associating with phrases like "poorly acted, poorly filmed, poorly written." These prototypes provide



Yelp

Prototype Aligned Interpretation		
	top-3 representative candidates	Contribution to Positive Class
Prototype 0	1. Yeah, the bar area is nice and colorful. 2. The inside of the location is decent, it mostly has tables a few booths and a new expanded bar. 3. It is always clean, outside patio with mist and a smoking section, TVs throughout the bar area.	0.797
Prototype 1	1. This place is better than you could imagine based on the concept and is well worth the meal. 2. The food is great, good portions and quality, yummy selections. 3. Not only is the food great, the service is impeccable.	0.765
Prototype 2	1. The food is horrible the service was bad. 2. The food was terrible and I would not recommend this place to anybody. 3. This is the WORST restaurant I have EVER been to and experienced.	-0.686
Prototype 3	1. When I came back to ask for a refund they were very rude about it and refused to help. 2. They then got my order wrong and said they wouldn't do anything about it when I told them. 3. I asked for a refund from the company and not a peep out of them.	-0.388
Prototype 4	1. I have to say I was truly disappointed by the flavors. 2. The flavors are \ok\, nothing special. 3. The only flavor I didn't care for was the red velvet.	-0.782
Prototype 5	1. Oh and the noise level was too high. 2. the place looked dirty and disorganized and smelled bad! 3. There was a dingy smell and a security guard wandering the aisles...	-0.433
Prototype 6	1. Service was good, friendly staff. 2. Service was excellent, staff was friendly. 3. The staff was nice and service was prompt.	0.517
Prototype 7	1. In addition to these items, the bread garlic butter that is served with the meal was also great. 2. We each ordered something different off the menu and everything was just scrumptious! 3. We were teated to a nice spread including vegetarian pasta primavera spicy!	0.432
Prototype 8	1. I am NEVER going back here and wouldn't recommend anyone to even try the place. 2. I wasn't impressed by this place and I don't think I'll be returning anytime soon. 3. I will never come here again.	-0.364
Prototype 9	1. Great price at $5.95 and plenty of food for me. 2. Prices are so reasonable, and with a restaurant.com coupon it was just dirt cheap. 3. The prices are insanely cheap for what you get.	0.481

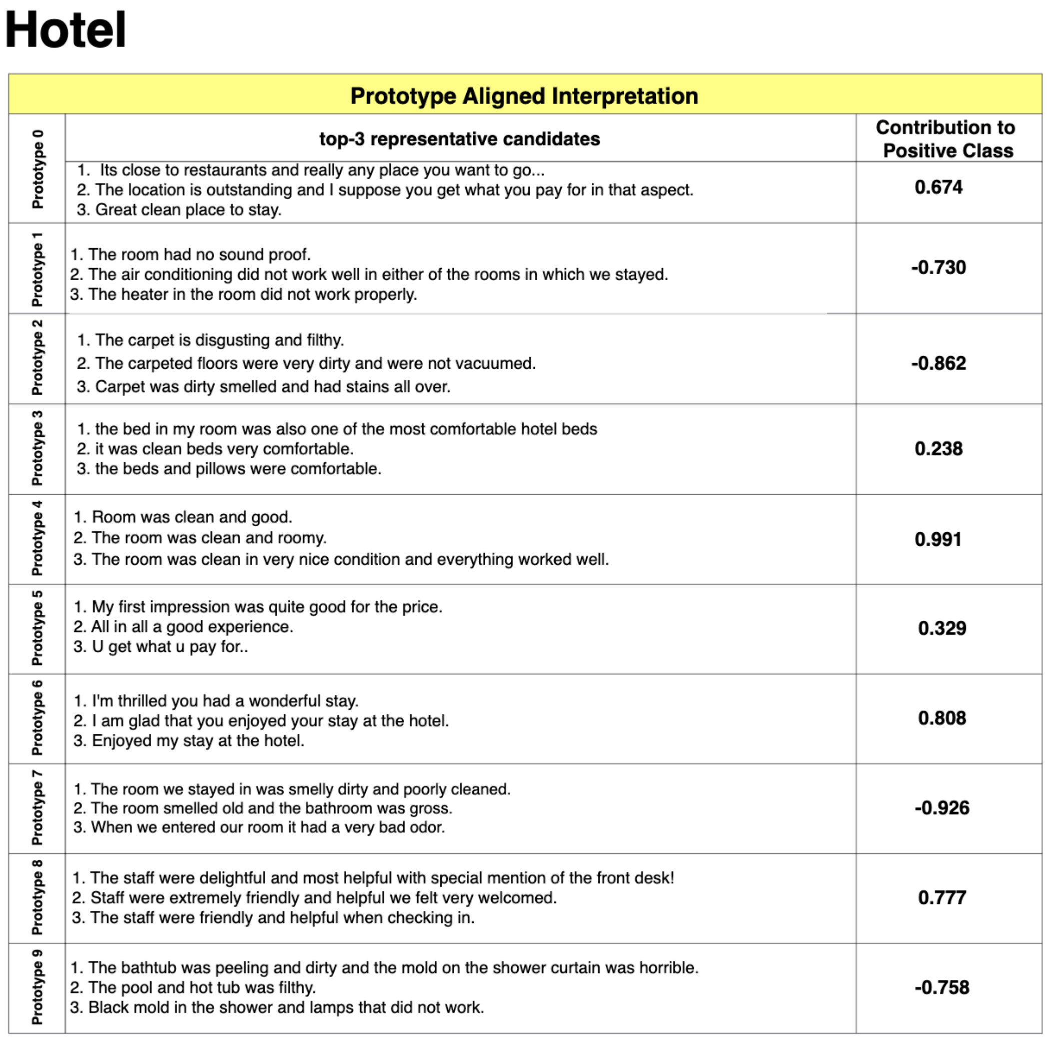
Figure 5: Aligned interpretation of prototypes with corresponding text sentences on the Yelp dataset. The figure highlights the diverse prototypes and their representative candidates, emphasizing interpretability in the sentiment analysis task.

interpretability by grounding the model's negative classification in representative samples that closely match the input text.

H.3 Interpretability

The examples in Figures 10 and 12 demonstrate ProtoLens' ability to explain its predictions using interpretable prototypes. By aligning input text with training set samples that serve as prototypes, ProtoLens offers a transparent view of how classification decisions are made. The similarity scores and sentiment weights ensure that each activated prototype meaningfully contributes to the overall prediction, enhancing both interpretability and faithfulness of the model.

Overall, these results highlight ProtoLens' capacity to provide human-understandable explanations for sentiment classification tasks, bridging the gap between model interpretability and practical applications.



Hotel

Prototype Aligned Interpretation		
Prototype 0	top-3 representative candidates	Contribution to Positive Class
	1. Its close to restaurants and really any place you want to go... 2. The location is outstanding and I suppose you get what you pay for in that aspect. 3. Great clean place to stay.	0.674
Prototype 1	1. The room had no sound proof. 2. The air conditioning did not work well in either of the rooms in which we stayed. 3. The heater in the room did not work properly.	-0.730
Prototype 2	1. The carpet is disgusting and filthy. 2. The carpeted floors were very dirty and were not vacuumed. 3. Carpet was dirty smelled and had stains all over.	-0.862
Prototype 3	1. the bed in my room was also one of the most comfortable hotel beds 2. it was clean beds very comfortable. 3. the beds and pillows were comfortable.	0.238
Prototype 4	1. Room was clean and good. 2. The room was clean and roomy. 3. The room was clean in very nice condition and everything worked well.	0.991
Prototype 5	1. My first impression was quite good for the price. 2. All in all a good experience. 3. U get what u pay for..	0.329
Prototype 6	1. I'm thrilled you had a wonderful stay. 2. I am glad that you enjoyed your stay at the hotel. 3. Enjoyed my stay at the hotel.	0.808
Prototype 7	1. The room we stayed in was smelly dirty and poorly cleaned. 2. The room smelled old and the bathroom was gross. 3. When we entered our room it had a very bad odor.	-0.926
Prototype 8	1. The staff were delightful and most helpful with special mention of the front desk! 2. Staff were extremely friendly and helpful we felt very welcomed. 3. The staff were friendly and helpful when checking in.	0.777
Prototype 9	1. The bathtub was peeling and dirty and the mold on the shower curtain was horrible. 2. The pool and hot tub was filthy. 3. Black mold in the shower and lamps that did not work.	-0.758

Figure 6: Aligned interpretation of prototypes with corresponding text sentences on the Hotel dataset. The interpretations include both positive and negative sentiment examples, showcasing the model's ability to capture nuanced feedback.

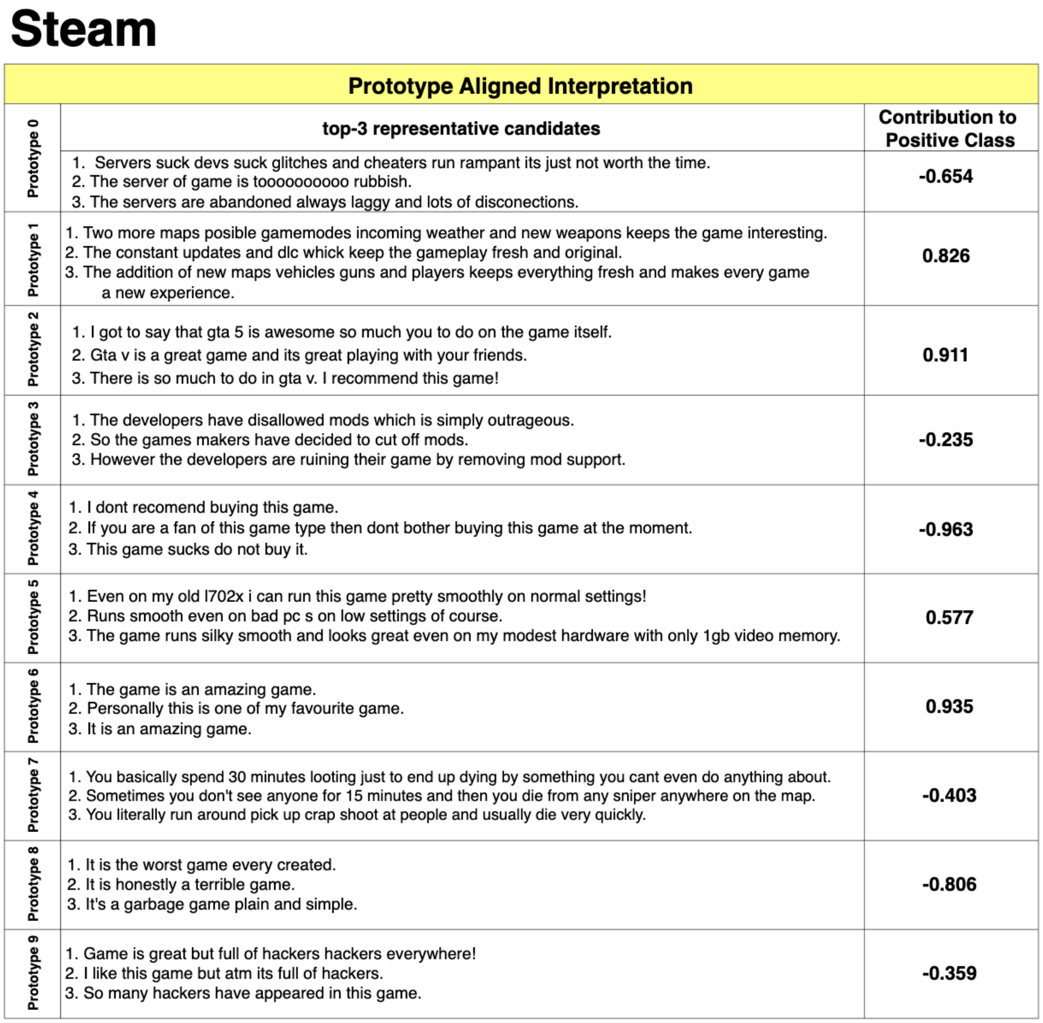
Table 2: Performance of ProtoLens with different ablation settings on various datasets.

Dataset	ProtoLens	*w/o Diversity*	*w/o Alignment*
IMDB	0.903	0.882	0.886
Amazon	0.937	0.926	0.927
Yelp	0.962	0.931	0.943
Hotel	0.963	0.947	0.953
Steam	0.931	0.917	0.923

I Ablation Study

To demonstrate the effectiveness of the Prototype Alignment and Diversity Constraint, we compare ProtoLens trained with and without these components. Prototype Alignment ensures that prototypes maintain their semantic faithfulness. The Diversity Constraint encourages prototypes to capture distinct, non-redundant features, enhancing generalization and reducing redundancy in representation. The results are shown in Table 2.

Impact of Diversity Constraints. The removal of diversity constraints (*w/o Diversity*) leads to a noticeable accuracy decline across all tested datasets, notably on IMDB (from 0.903 to 0.882),



Steam

Prototype Aligned Interpretation		
	top-3 representative candidates	Contribution to Positive Class
Prototype 0	1. Servers suck devs suck glitches and cheaters run rampant its just not worth the time. 2. The server of game is tooooooooo rubbish. 3. The servers are abandoned always laggy and lots of disconections.	-0.654
Prototype 1	1. Two more maps posible gamemodes incoming weather and new weapons keeps the game interesting. 2. The constant updates and dlc whick keep the gameplay fresh and original. 3. The addition of new maps vehicles guns and players keeps everything fresh and makes every game a new experience.	0.826
Prototype 2	1. I got to say that gta 5 is awesome so much you to do on the game itself. 2. Gta v is a great game and its great playing with your friends. 3. There is so much to do in gta v. I recommend this game!	0.911
Prototype 3	1. The developers have disallowed mods which is simply outrageous. 2. So the games makers have decided to cut off mods. 3. However the developers are ruining their game by removing mod support.	-0.235
Prototype 4	1. I dont recomend buying this game. 2. If you are a fan of this game type then dont bother buying this game at the moment. 3. This game sucks do not buy it.	-0.963
Prototype 5	1. Even on my old l702x i can run this game pretty smoothly on normal settings! 2. Runs smooth even on bad pc s on low settings of course. 3. The game runs silky smooth and looks great even on my modest hardware with only 1gb video memory.	0.577
Prototype 6	1. The game is an amazing game. 2. Personally this is one of my favourite game. 3. It is an amazing game.	0.935
Prototype 7	1. You basically spend 30 minutes looting just to end up dying by something you cant even do anything about. 2. Sometimes you don't see anyone for 15 minutes and then you die from any sniper anywhere on the map. 3. You literally run around pick up crap shoot at people and usually die very quickly.	-0.403
Prototype 8	1. It is the worst game every created. 2. It is honestly a terrible game. 3. It's a garbage game plain and simple.	-0.806
Prototype 9	1. Game is great but full of hackers hackers everywhere! 2. I like this game but atm its full of hackers. 3. So many hackers have appeared in this game.	-0.359

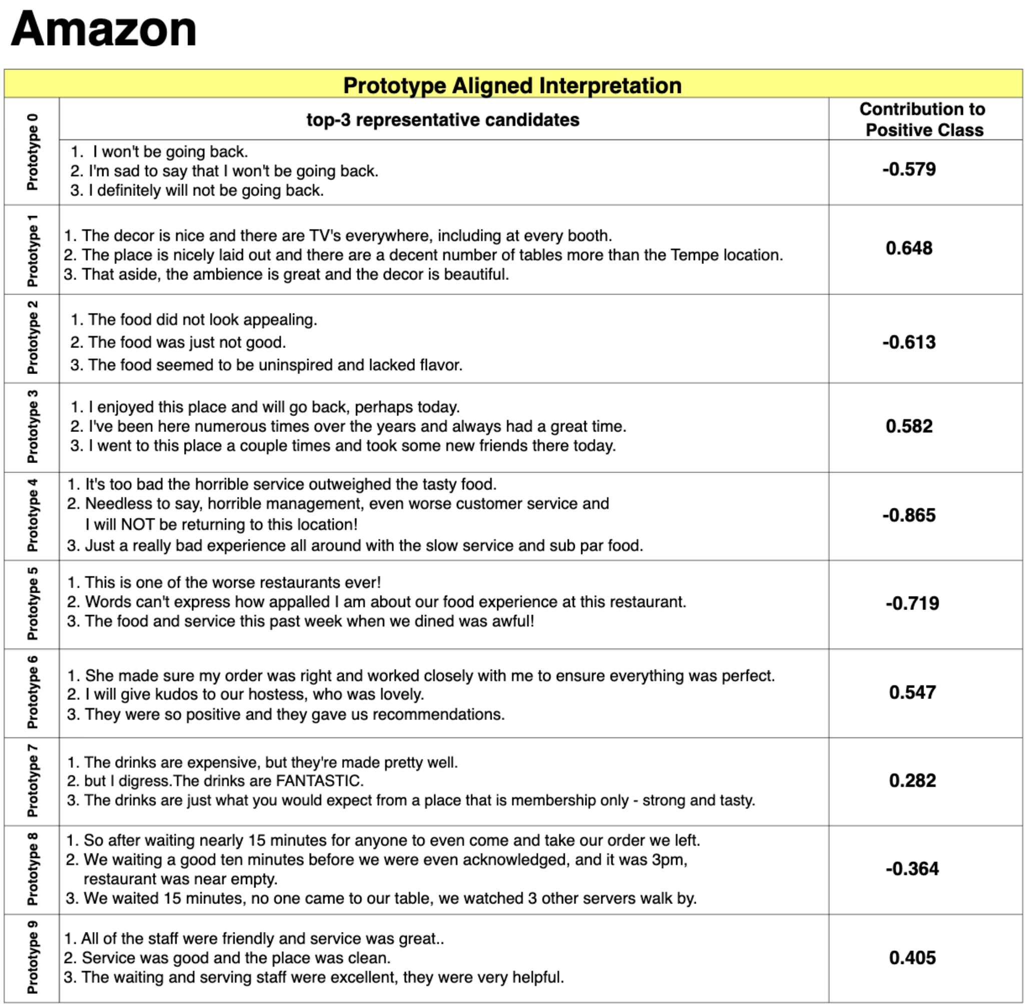
Figure 7: Aligned interpretation of prototypes with corresponding text sentences on the Steam dataset. The figure demonstrates how ProtoLens handles diverse feedback in gaming reviews, including issues like performance and user experience.

Amazon (from 0.937 to 0.926), Yelp (from 0.962 to 0.931) and Hotel (from 0.963 to 0.947). This indicates that the diversity loss plays a crucial role in encouraging distinct and varied prototype representations, which helps the model generalize better across different data points. The drop in accuracy suggests that when prototypes become more redundant, they lose their ability to represent the diversity in the dataset, limiting the model's interpretability and performance.

Impact of Prototype Alignment. The ablation results for removing prototype alignment (*w/o Alignment*) show a decline in performance, particularly on the Yelp dataset (from 0.963 to 0.943), highlighting the importance of prototype alignment. Aligning prototypes with representative embeddings ensures they remain semantically meaningful, leading to more accurate and interpretable predictions. The slight performance drop across other datasets, such as IMDB and Amazon, further emphasizes that the adaptive update process enabled by prototype alignment promotes more stable and reliable learning, improving the model's interpretability and accuracy.

J Hyperparameter

Effect of K. The number of prototypes, denoted by K, plays a crucial role in determining the balance between model interpretability and classification performance. As shown in Figure 13, increasing



Amazon

Prototype Aligned Interpretation

	top-3 representative candidates	Contribution to Positive Class
Prototype 0	1. I won't be going back. 2. I'm sad to say that I won't be going back. 3. I definitely will not be going back.	-0.579
Prototype 1	1. The decor is nice and there are TV's everywhere, including at every booth. 2. The place is nicely laid out and there are a decent number of tables more than the Tempe location. 3. That aside, the ambience is great and the decor is beautiful.	0.648
Prototype 2	1. The food did not look appealing. 2. The food was just not good. 3. The food seemed to be uninspired and lacked flavor.	-0.613
Prototype 3	1. I enjoyed this place and will go back, perhaps today. 2. I've been here numerous times over the years and always had a great time. 3. I went to this place a couple times and took some new friends there today.	0.582
Prototype 4	1. It's too bad the horrible service outweighed the tasty food. 2. Needless to say, horrible management, even worse customer service and I will NOT be returning to this location! 3. Just a really bad experience all around with the slow service and sub par food.	-0.865
Prototype 5	1. This is one of the worse restaurants ever! 2. Words can't express how appalled I am about our food experience at this restaurant. 3. The food and service this past week when we dined was awful!	-0.719
Prototype 6	1. She made sure my order was right and worked closely with me to ensure everything was perfect. 2. I will give kudos to our hostess, who was lovely. 3. They were so positive and they gave us recommendations.	0.547
Prototype 7	1. The drinks are expensive, but they're made pretty well. 2. but I digress.The drinks are FANTASTIC. 3. The drinks are just what you would expect from a place that is membership only - strong and tasty.	0.282
Prototype 8	1. So after waiting nearly 15 minutes for anyone to even come and take our order we left. 2. We waiting a good ten minutes before we were even acknowledged, and it was 3pm, restaurant was near empty. 3. We waited 15 minutes, no one came to our table, we watched 3 other servers walk by.	-0.364
Prototype 9	1. All of the staff were friendly and service was great.. 2. Service was good and the place was clean. 3. The waiting and serving staff were excellent, they were very helpful.	0.405

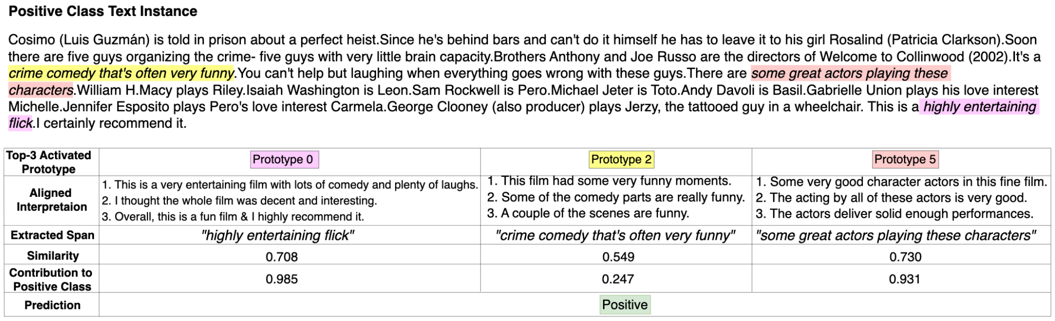
Figure 8: Aligned interpretation of prototypes with corresponding text sentences on the Amazon dataset. This figure illustrates ProtoLens' interpretability across product reviews, focusing on features such as quality, service, and usability.

K generally leads to improved accuracy across most datasets, with the exception of some slight fluctuations. For instance, in the IMDB dataset, increasing K from 10 to 40 boosts the performance from 0.884 to 0.903, while for the Yelp dataset, a similar increase elevates the accuracy from 0.931 to 0.950. The improvements plateau or slightly decrease for higher values of K, suggesting diminishing returns beyond a certain point.

The optimal value of K appears to be dataset-dependent. For example, $K = 50$ yields the best performance on the Amazon and Yelp datasets with 0.937 and 0.962, respectively, while $K = 40$ provides the best performance on the IMDB dataset (0.903). Meanwhile, for the Hotel dataset, $K = 20$ achieves the highest accuracy at 0.963. This variation indicates that the ideal number of prototypes may depend on the complexity and size of the dataset.

Overall, increasing K allows the model to capture more fine-grained patterns by using a larger set of prototypes, but setting K too high may introduce unnecessary complexity without substantial accuracy gains. Thus, choosing K involves a trade-off between maintaining a manageable number of interpretable prototypes and achieving high predictive performance.

Effect of n-gram. An n-gram is a hyperparameter that determines the granularity of text division. As shown in Figure 14, an n-gram size of 5 achieves the best performance across all datasets, with notable improvements on IMDB (0.903), Amazon (0.937), and Hotel (0.963), indicating that $n = 5$

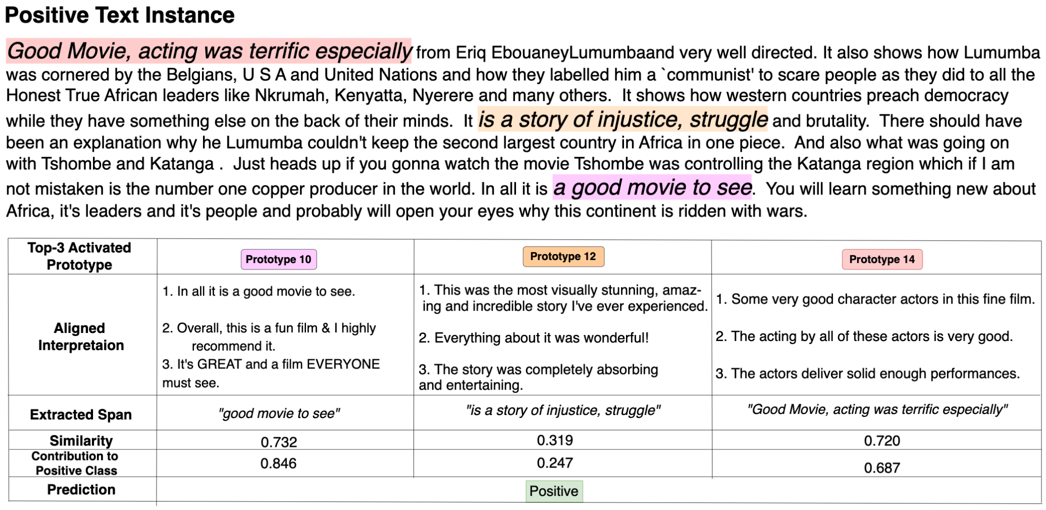


Positive Class Text Instance

Cosimo (Luis Guzmán) is told in prison about a perfect heist.Since he's behind bars and can't do it himself he has to leave it to his girl Rosalind (Patricia Clarkson).Soon there are five guys organizing the crime- five guys with very little brain capacity.Brothers Anthony and Joe Russo are the directors of Welcome to Collinwood (2002).It's a *crime comedy that's often very funny*.You can't help but laughing when everything goes wrong with these guys.There are *some great actors playing these characters*.William H.Macy plays Riley.Isaiah Washington is Leon.Sam Rockwell is Pero.Michael Jeter is Toto.Andy Davoli is Basil.Gabrielle Union plays his love interest Michelle.Jennifer Esposito plays Pero's love interest Carmela.George Clooney (also producer) plays Jerzy, the tattooed guy in a wheelchair. This is a *highly entertaining flick*.I certainly recommend it.

Top-3 Activated Prototype	Prototype 0	Prototype 2	Prototype 5
Aligned Interpretaion	1. This is a very entertaining film with lots of comedy and plenty of laughs. 2. I thought the whole film was decent and interesting. 3. Overall, this is a fun film & I highly recommend it.	1. This film had some very funny moments. 2. Some of the comedy parts are really funny. 3. A couple of the scenes are funny.	1. Some very good character actors in this fine film. 2. The acting by all of these actors is very good. 3. The actors deliver solid enough performances.
Extracted Span	*"highly entertaining flick"*	*"crime comedy that's often very funny"*	*"some great actors playing these characters"*
Similarity	0.708	0.549	0.730
Contribution to Positive Class	0.985	0.247	0.931
Prediction		Positive	

Figure 9: Case study of a positive class text instance.ProtoLens identifies relevant prototypes (e.g., "highly entertaining flick") and aligns them with specific spans in the input text. Extracted spans, similarity scores, and sentiment weights show how each prototype contributes to the positive prediction.



Positive Text Instance

Good Movie, acting was terrific especially from Eriq EbouaneyLumumbaand very well directed. It also shows how Lumumba was cornered by the Belgians, U S A and United Nations and how they labelled him a `communist' to scare people as they did to all the Honest True African leaders like Nkrumah, Kenyatta, Nyerere and many others. It shows how western countries preach democracy while they have something else on the back of their minds. It *is a story of injustice, struggle* and brutality. There should have been an explanation why he Lumumba couldn't keep the second largest country in Africa in one piece. And also what was going on with Tshombe and Katanga . Just heads up if you gonna watch the movie Tshombe was controlling the Katanga region which if I am not mistaken is the number one copper producer in the world. In all it is *a good movie to see*. You will learn something new about Africa, it's leaders and it's people and probably will open your eyes why this continent is ridden with wars.

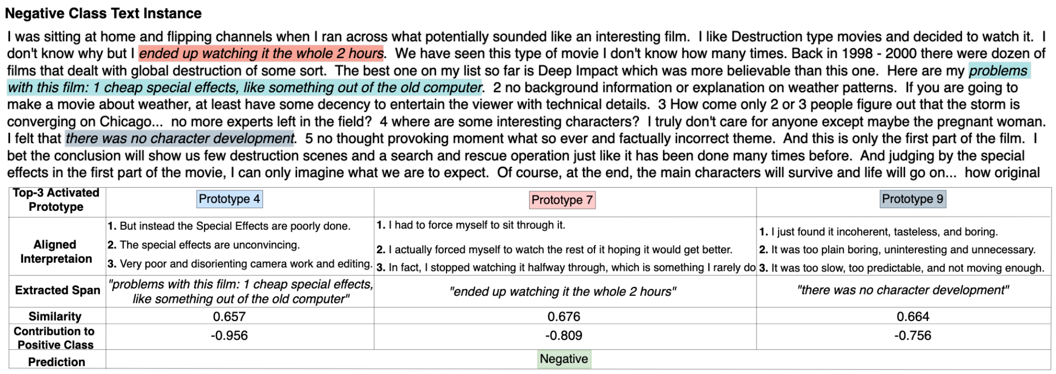
Top-3 Activated Prototype	Prototype 10	Prototype 12	Prototype 14
Aligned Interpretaion	1. In all it is a good movie to see. 2. Overall, this is a fun film & I highly recommend it. 3. It's GREAT and a film EVERYONE must see.	1. This was the most visually stunning, amazing and incredible story I've ever experienced. 2. Everything about it was wonderful! 3. The story was completely absorbing and entertaining.	1. Some very good character actors in this fine film. 2. The acting by all of these actors is very good. 3. The actors deliver solid enough performances.
Extracted Span	*"good movie to see"*	*"is a story of injustice, struggle"*	*"Good Movie, acting was terrific especially"*
Similarity	0.732	0.319	0.720
Contribution to Positive Class	0.846	0.247	0.687
Prediction		Positive	

Figure 10: The figure showcases how ProtoLens aligns input text with prototypes to explain a positive sentiment prediction. The extracted spans and similarities for the top-3 activated prototypes are presented, along with sentiment weights contributing to the final prediction.

is the optimal n-gram size, providing the best trade-off between incorporating sufficient context and avoiding unnecessary complexity. For smaller n-gram sizes (e.g., $n = 1, 3$), performance is slightly lower, likely due to the model's limited ability to capture broader contextual information. On the other hand, a larger n-gram size ($n = 7, 9$) does not yield improved performance and even leads to a decrease in accuracy on all datasets, as seen with IMDB and Amazon. This suggests that including too large of a n-gram introduces noise, which results in slight performance degradation.

K Impact of Encoder

To further assess generalizability, we evaluate ProtoLens using alternative pre-trained encoders, including T5 and bge-m3, while keeping all other settings fixed. Results in Table 3 confirm that ProtoLens remains effective across architectures. Notably, the bge-m3 encoder yields the highest accuracy on all three datasets, surpassing MPNet. ProtoLens with T5 performs comparably, with only minor degradation, likely due to differences in embedding granularity. These results show that ProtoLens is not tied to a specific encoder and can flexibly adapt to various pretrained language models, reinforcing its scalability and broad applicability.

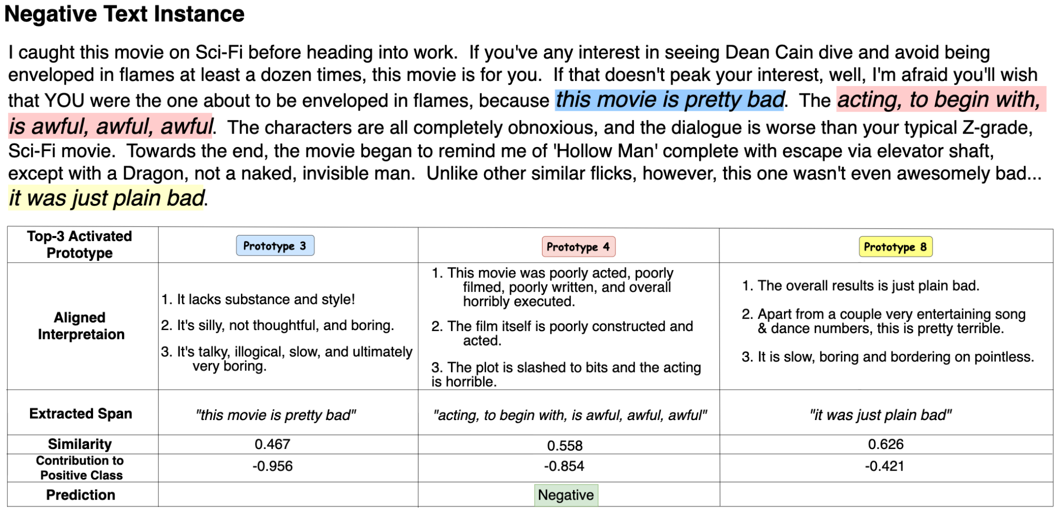


Negative Class Text Instance

I was sitting at home and flipping channels when I ran across what potentially sounded like an interesting film. I like Destruction type movies and decided to watch it. I don't know why but I *ended up watching it the whole 2 hours*. We have seen this type of movie I don't know how many times. Back in 1998 - 2000 there were dozen of films that dealt with global destruction of some sort. The best one on my list so far is Deep Impact which was more believable than this one. Here are my *problems with this film: 1 cheap special effects, like something out of the old computer*. 2 no background information or explanation on weather patterns. If you are going to make a movie about weather, at least have some decency to entertain the viewer with technical details. 3 How come only 2 or 3 people figure out that the storm is converging on Chicago... no more experts left in the field? 4 where are some interesting characters? I truly don't care for anyone except maybe the pregnant woman. I felt that *there was no character development*. 5 no thought provoking moment what so ever and factually incorrect theme. And this is only the first part of the film. I bet the conclusion will show us few destruction scenes and a search and rescue operation just like it has been done many times before. And judging by the special effects in the first part of the movie, I can only imagine what we are to expect. Of course, at the end, the main characters will survive and life will go on... how original

Top-3 Activated Prototype	Prototype 4	Prototype 7	Prototype 9
Aligned Interpretaion	**1.** But instead the Special Effects are poorly done. **2.** The special effects are unconvincing. **3.** Very poor and disorienting camera work and editing.	**1.** I had to force myself to sit through it. **2.** I actually forced myself to watch the rest of it hoping it would get better. **3.** In fact, I stopped watching it halfway through, which is something I rarely do	**1.** I just found it incoherent, tasteless, and boring. **2.** It was too plain boring, uninteresting and unnecessary. **3.** It was too slow, too predictable, and not moving enough.
Extracted Span	*"problems with this film: 1 cheap special effects, like something out of the old computer"*	*"ended up watching it the whole 2 hours"*	*"there was no character development"*
Similarity	0.657	0.676	0.664
Contribution to Positive Class	-0.956	-0.809	-0.756
Prediction		Negative	

Figure 11: Case study of a negative class text instance. ProtoLens identifies relevant prototypes (e.g., "there was no character development") and aligns them with specific spans in the input text. Extracted spans, similarity scores, and sentiment weights show how each prototype contributes to the negative prediction.



Negative Text Instance

I caught this movie on Sci-Fi before heading into work. If you've any interest in seeing Dean Cain dive and avoid being enveloped in flames at least a dozen times, this movie is for you. If that doesn't peak your interest, well, I'm afraid you'll wish that YOU were the one about to be enveloped in flames, because *this movie is pretty bad*. The *acting, to begin with, is awful, awful, awful*. The characters are all completely obnoxious, and the dialogue is worse than your typical Z-grade, Sci-Fi movie. Towards the end, the movie began to remind me of 'Hollow Man' complete with escape via elevator shaft, except with a Dragon, not a naked, invisible man. Unlike other similar flicks, however, this one wasn't even awesomely bad... *it was just plain bad*.

Top-3 Activated Prototype	Prototype 3	Prototype 4	Prototype 8
Aligned Interpretaion	1. It lacks substance and style! 2. It's silly, not thoughtful, and boring. 3. It's talky, illogical, slow, and ultimately very boring.	1. This movie was poorly acted, poorly filmed, poorly written, and overall horribly executed. 2. The film itself is poorly constructed and acted. 3. The plot is slashed to bits and the acting is horrible.	1. The overall results is just plain bad. 2. Apart from a couple very entertaining song & dance numbers, this is pretty terrible. 3. It is slow, boring and bordering on pointless.
Extracted Span	*"this movie is pretty bad"*	*"acting, to begin with, is awful, awful, awful"*	*"it was just plain bad"*
Similarity	0.467	0.558	0.626
Contribution to Positive Class	-0.956	-0.854	-0.421
Prediction		Negative	

Figure 12: The figure shows how ProtoLens aligns input text with prototypes to explain a negative sentiment prediction, supported by similarity scores and sentiment weights.

L Cross-Dataset Prototype Generalization

To assess the generalizability of ProtoLens prototypes across datasets, we conducted a cross-dataset evaluation. Specifically, we tested the performance of ProtoLens on the Hotel dataset using prototypes derived from the Yelp and Amazon datasets, which also represent customer review domains. Table 4 summarizes the results.

The results demonstrate that ProtoLens maintains strong performance even when using prototypes derived from external datasets. While the accuracy slightly decreases compared to using prototypes generated directly from the target dataset (Hotel), the drop in performance is modest: a 0.9% and 2.0% reduction in accuracy when using Yelp and Amazon prototypes, respectively. This suggests that ProtoLens prototypes capture generalizable patterns that can extend across datasets with similar domains.

These findings underscore the robustness of ProtoLens in leveraging prototypes across related datasets, a desirable property for practical applications where annotated data for prototype derivation may be limited. Furthermore, the ability to generalize across datasets indicates that ProtoLens can identify domain-invariant concepts, making it a promising approach for transfer learning and cross-domain interpretability in prototype-based models.

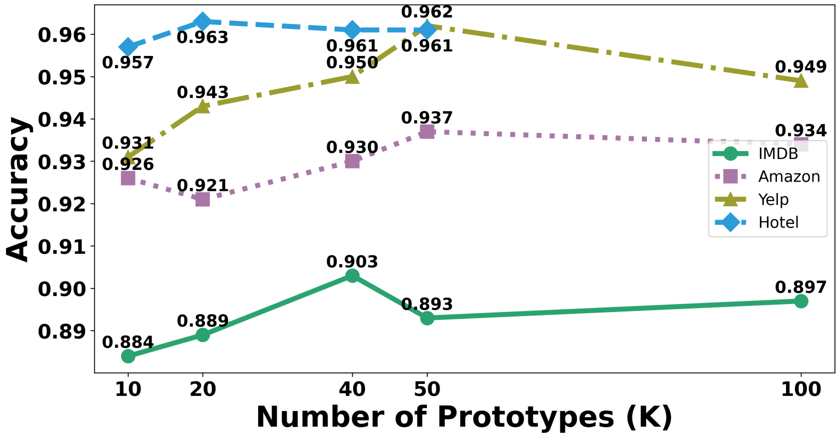


Figure 13: Performance of ProtoLens in comparison with different number of prototypes. Performance improves with more prototypes, peaking at an optimal K (e.g., 40 for IMDB), before stabilizing or slightly decreasing.

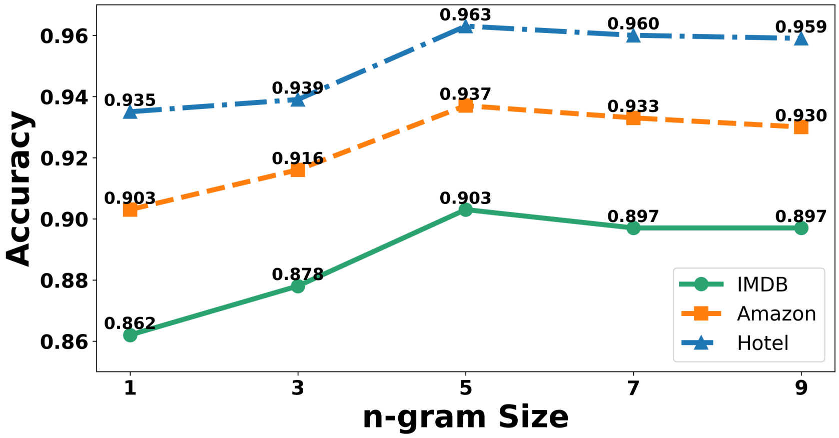


Figure 14: Accuracy of ProtoLens across IMDB, Amazon, and Hotel datasets as n-gram size varies. Larger n-grams improve contextual representation, but performance plateaus or slightly decreases beyond n=5, indicating a tradeoff between context and generalizability.

M LLaMA-3 Prompt

We use the following zero-shot prompt for LLaMA-3-8B in all experiments:

```
Given a movie review, your job is to classify its sentiment
into binary class: positive or negative.
Review: {input_text}
Do not provide reasoning or explanation. Output should be one
word only: "Negative" or "Positive".
sentiment:
```

Dataset	**MPNet**	**T5**	**bge-m3**
Hotel	0.963	0.955	**0.968**
Amazon	0.937	0.909	**0.942**
IMDB	0.903	0.897	**0.924**

Table 3: Performance of ProtoLens with different foundation models on three benchmark datasets.

Table 4: Cross-dataset evaluation results. ProtoLens performance on the Hotel dataset with prototypes derived from different datasets.

Prototype Source	**Accuracy on Hotel Dataset**
Hotel (Original)	0.963
Yelp	0.954
Amazon	0.943