

Progressive Screenplay Narrative Understanding via Contrastive Learning

Stanford CS224N Custom Project

Luca Thomas Wheeler

Department of Computer Science
Stanford University
ltw@stanford.edu

Abstract

Screenplays contain an implicit narrative structure that often spans 100+ scenes, however standard embedding models encode each scene independently, discarding sequential context. We train a causal transformer with an InfoNCE contrastive objective to autoregressively contextualize scene embeddings, learning which scenes coherently follow from the narrative so far at any point in the screenplay. Our 1.4M-parameter model nearly matches a strong copy baseline on next-scene ranking (0.532 vs. 0.538 Hits@1) while **outperforming** it on longer contexts (78.6% vs. 64.3% at position 75) and achieving 0.942 AUC on scene-intruder detection. Linear probing confirms the model learns narrative structure: contextualized embeddings achieve $R^2 = 0.668$ for position regression versus $R^2 = 0.092$ for raw embeddings and **97.8% accuracy** on first/last act discrimination vs. 67.6%.

1 Key Information to include

- Mentor: **Nevin George**
- External Collaborators (if you have any): **N/A**
- Sharing project: **N/A**

2 Introduction

Screenplays are some of the longest and most structurally complicated documents in NLP. One standard script could have 100+ scenes and multiple storylines facilitating turning points, escalations, and resolutions never formally defined. Most prior computational work on screenplays has focused on character analysis [1], social networks [2], and summarization [3], and direct attempts at modeling narrative structure [4] depend heavily on human-annotated synopses that try to binarize a fluid art-form. A core question remains: can we learn continuous representations of narrative progression without labeled structural data?

Standard embedding models encode scenes independent of one another. This process discards the sequential aspect that is the basis of narrative meaning. A wave in scene one means something very different than in scene one-hundred, but a contextless language encoder sees the same thing. We seek to answer the question of whether self-supervision and a contrastive objective can recover the missing context, carrying the weight of a wave goodbye. We train a causal transformer, our CausalSceneTransformer, that reads scene embeddings in order and learns to rank the correct next scene above negatives via InfoNCE loss, forcing the model to build internal representations of narrative state that go beyond static scene content.

Our 1.4M-parameter model nearly matches a strong copy baseline on next-scene ranking (0.532 vs. 0.538 Hits@1) while outperforming it on calibration (0.739 vs. 0.698 Hits@3) and on longer contexts (78.6% vs. 64.3% Hits@1 at position 75), suggesting the transformer learns to leverage

extended narrative arcs that surface-similarity baselines cannot capture. Linear probing confirms this: contextualized embeddings achieve $R^2 = 0.668$ for narrative position regression versus $R^2 = 0.092$ for raw embeddings and 97.8% accuracy on first/last act discrimination versus 67.6%. What that means: meaningful structural information emerges as a byproduct of next-scene prediction, without any supervision or labeled data.

3 Related Work

Narrative and screenplay understanding: Computational work on screenplays has previously focused on characters rather than story structure: Bamman et al. learn character types from plot summaries [1], Chaturvedi et al. model inter-character relationships [5], and Agarwal et al. extract social networks using scripts [2]. Gorinski and Lapata [3] do tackle screenplay summarization, however they rely on surface features rather than narrative progression. The most direct attempt at modeling narrative structure is Papalampidi et al. [4], who formalized turning point (TP) identification as an NLP task, published the TRIPOD dataset (99 films with TP annotations), and presented a Topic-Aware Model (TAM) that contextualizes sentence embeddings with a BiLSTM and a context interaction layer inspired by work in text segmentation [6]. Their best model achieves 41.33% total agreement on synopsis TPs versus 28% for a position baseline. It’s important to remember: TAM requires a human-written plot synopsis to analyze a screenplay and treats TPs as discrete binary labels, which discards the continuous nature of narrative progression. Follow-up work [7] shows that structural understanding does improve downstream summarization, which implies narrative structure is a practical bottleneck rather than just an intellectual curiosity. We include TRIPOD as an evaluation benchmark but approach the problem from a different route: learning continuous narrative representations through self-supervision rather than supervised classification.

Document structure and long-form understanding: The core NLP problem we address in this paper is representing long documents with implicit internal structure. Document structure understanding has been studied for scientific papers [8], where sections like Abstract/Methods/Results provide explicit markers, and for news articles which leverage the inverted pyramid convention. Creative narratives, however, have been, for the most part, ignored. Screenplays occupy a unique middle ground: they have explicit formatting conventions (slug lines, scene-cuts, block-types) but implicit narrative structure (inciting incidents, turning points, climaxes) that never get marked in the text, but comes through in subtext and intuition. Hearst’s TextTiling [6] segments documents by topical coherence and subject changes (the TP framework applies a similar idea to story beats), but both assume that structural boundaries are local phenomena. Our approach instead models structure as a progressive, global property dependent on sequential-context.

Contrastive learning for representations: Our training objective builds on Contrastive Predictive Coding (CPC) [9], which uses an InfoNCE loss to learn representations by predicting future observations from context. This framework has been applied to speech, vision, and text, demonstrating that contrastive self-supervision can recover latent structure without labels. In NLP, contrastive objectives are the basis of sentence-level representation learning [10] and have been shown capable of learning document-level structure. We apply the same principle at the scene level: rather than predicting words or sentences, our model predicts which scene naturally follows from the narrative so far, using a contrastive signal to build representations for narrative state.

4 Approach

We parsed screenplays from the IMSDB archive into scene-level text, truncating each scene to 2,000 characters, and embedded them using a frozen all-MiniLM-L6-v2 encoder [10], giving us 384-dimensional embeddings. We then input these static embeddings to our **CausalSceneTransformer**, our original architecture which contextualizes a scene’s representation autoregressively. We use a linear projection to map each embedding to d_{model} dimensions before adding learned positional embeddings and passing the result through a causal transformer. At position t the model attends only to scenes $1, \dots, t$ and produces a contextualized representation h_t . An output projection (Linear to GELU to Linear) maps h_t back to the 384-dimensional input space for direct comparison with raw scene embeddings. In training we use InfoNCE loss [9]:

$$\mathcal{L} = -\log \frac{\exp(\text{sim}(h_t, e_{t+1})/\tau)}{\sum_{j=1}^K \exp(\text{sim}(h_t, e_j)/\tau)} \quad (1)$$

where sim is cosine similarity and e_{t+1} is the ground-truth next scene’s raw embedding. τ is a fixed temperature of 0.07 and $K=32$ candidates includes the true next scene and 31 negatives with roughly a third sampled from within the same screenplay and the rest from others in the training set.

Our baselines are: **Copy**: rank by cosine similarity to scene t ’s raw embedding. **Mean**: cosine similarity to the running average of all prior scene embeddings. **Position-only**: the trained model with zeroed-out scene content, so just our positional embeddings. **Random**: uniform ranking.

5 Experiments

5.1 Data

Originally, we tried to scrape the full 1,302 screenplays from IMSDB, but only 824 were successfully parsed into scenes (76 fetch failures, 402 had too few scenes). Our final corpus is 823 screenplays with 111,154 total scenes (splitting on every slug-line), each embedded. The average screenplay contains 135.1 scenes (std 70.2, min 5, max 616). We split our dataset at the screenplay level: 658 train, 82 dev, 83 test. We trained in two rounds: Round 1 used 149 screenplays to validate the pipeline but showed severe overfitting, motivating Round 2 scaling to our full corpus.

5.2 Evaluation method

We evaluate on four fronts. 1) *Next-scene ranking*: Hits@1, Hits@3, and MRR over 64 seeded candidates per example. 2) *Scene-intruder detection*: AUC when replacing a random scene with one from a different screenplay. 3) *Context-length analysis*: Hits@1 measured at specific narrative positions (2, 5, 10, 15, 20, 30, 50, 75, 100) to test whether performance improves with longer context. 4) *Linear probing*: we train linear classifiers on raw versus contextualized embeddings to predict narrative position (regression, quintile classification, decile classification) and first vs. last act discrimination. This tests whether the transformer adds structural information beyond what the frozen encoder provides.

5.3 Experimental details

We used MiniLM (384-dim) rather than a larger encoder so that learned structure is attributable to contextualization rather than richer input representations. We trained with AdamW (lr=1e-4, weight decay 0.01), cosine annealing (T_max=100, eta_min=1e-6), batch size 32, and gradient clipping at 1.0. We tested two model sizes. **Small**: 2 layers, 4 heads, $d_{\text{model}}=256$, $d_{\text{ff}}=512$, dropout 0.2 (1.4M params). **Large**: 4 layers, 8 heads, $d_{\text{model}}=512$, $d_{\text{ff}}=2048$, dropout 0.1 (13.4M params). Early stopping with patience 10 (eval every 5 epochs). The Small model peaked at epoch 10 and stopped at epoch 60; the Large peaked at epoch 5 and stopped at epoch 55.

5.4 Results

Table 1 reports our main findings on the 83-screenplay test set.

Model	Hits@1	Hits@3	MRR	Intruder AUC
Copy Baseline	0.538	0.698	0.640	–
Small (1.4M)	0.532	0.739	0.658	0.942
Mean Baseline	0.495	0.734	0.634	–
Large (13.4M)	0.467	0.684	0.602	0.919
Small-HardNeg (1.4M)	0.373	0.616	0.528	0.910
Position-only	0.016	0.051	0.078	–
Random	0.015	0.047	0.074	–

Table 1: Next-scene ranking and intruder detection on the test set

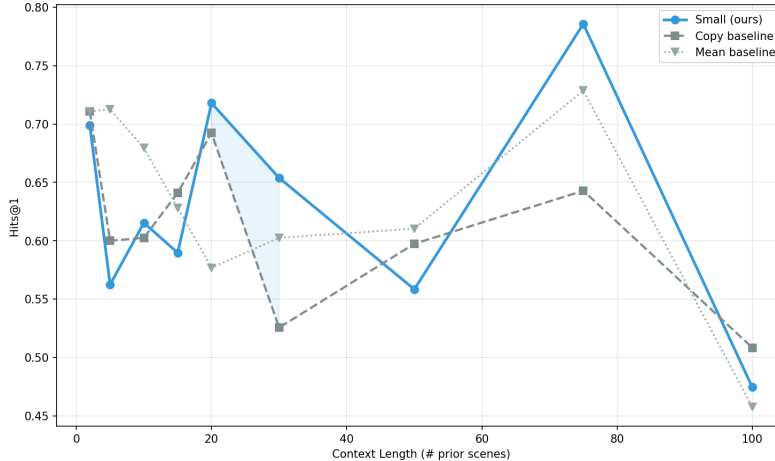


Figure 1: Hits@1 vs. context length

The Small model nearly matches Copy on Hits@1 (0.532 vs. 0.538) but outperforms it on Hits@3 (0.739 vs. 0.698) and MRR (0.658 vs. 0.640). This means its overall ranking is better organized so even when it misses on rank-1 the correct scene is usually still near the top. Copy’s Hits@1 strength confirms that adjacent scenes tend to share locations and characters [4], making surface similarity a strong heuristic. Position-only scores at random (0.016), confirming the model relies on scene content rather than positional statistics.

Data scaling yielded the single biggest improvement. Going from Round 1 (149 screenplays, Large model) to Round 2 (823 screenplays, Small model) improved Hits@1 from 0.193 to 0.532, largely eliminating overfitting. The Large model still overfits very quickly (best at epoch 5) while the Small model trains smooth to epoch 10 (best dev loss 2.47).

The context-length analysis in Table 2 demonstrates our transformer’s advantage. With short contexts (positions 2 through 10) Copy and Mean are competitive or stronger than our model, but the Small model pulls ahead at longer contexts, peaking at position 75 with 78.6% vs. 64.3% for Copy. This supports our central hypothesis: the transformer learns to integrate extended narrative context in ways single-scene similarity cannot.

Position	Small	Large	Copy	Mean	n
2	0.699	0.566	0.711	0.711	83
10	0.615	0.590	0.603	0.679	78
20	0.718	0.590	0.692	0.577	78
30	0.654	0.500	0.526	0.603	78
50	0.558	0.558	0.597	0.610	77
75	0.786	0.700	0.643	0.729	70
100	0.475	0.458	0.508	0.458	59

Table 2: Hits@1 by narrative position (number of prior scenes seen). n here is the number of test screenplays with at least that many scenes.

We believe the position 100 drop (0.475 to Copy’s 0.508) reflects selection bias: only 59 screenplays reach 100+ scenes, and these longer scripts may have different structural properties as discussed in Section 6.

In addition to these tests, we apply linear probing to evaluate what information our transformer adds. We trained linear classifiers on raw MiniLM embeddings versus contextualized embeddings from the Small model to predict narrative position. Table 3 reports the results.

Across every task our contextualized embeddings dramatically outperform raw embeddings. Position regression R^2 jumps from 0.092 to 0.668 and first/last act discrimination reaches 97.8%.

Task	Raw (MiniLM)	Contextualized
Position regression (R^2)	0.092	0.668
Quintile accuracy (5-class)	0.288	0.534
Decile accuracy (10-class)	0.161	0.366
First/last act accuracy	0.676	0.978

Table 3: Linear probing accuracy on raw vs. contextualized embeddings.

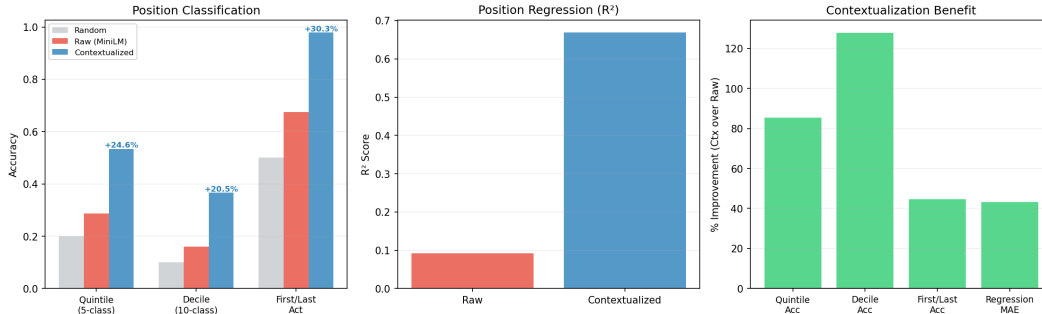


Figure 2: Linear Probing: Raw vs. Contextualized

6 Analysis

The linear probing results in Table 3 are our strongest evidence that the CausalSceneTransformer learns meaningful narrative structure. Raw MiniLM embeddings carry almost no positional information with an $R^2 = 0.092$ for position regression, barely above zero. This makes sense: a scene’s content (dialogue, action, etc.) tells us much less about where it falls in a story than where we’ve been. To that end, after contextualization R^2 jumps to 0.668. This means the model encodes and understands where we are in the story, despite never seeing any labels. First/last act discrimination goes from 67.6% to 97.8%, meaning the model nearly perfectly distinguishes early scenes from late ones.

It’s important to reiterate: this is not a result of positional embeddings leaking through. The Position-only ablation (0.016 Hits@1, effectively random) confirms that positional embeddings carry no useful signal without scene content, and the probed representations live in the output space after a nonlinear projection, not the positional embedding space directly. The structural information in these representations was all learned from the contrastive objective.

On the TRIPOD turning point correlation, we tested whether scenes where the model shows high “narrative surprise” (low next-scene ranking confidence) align with the human-annotated turning points from TRIPOD [4]. At the scene level, using the four films with gold-standard annotations, we found no significant signal (all $p > 0.39$). That said, this sample is small and makes drawing firm conclusions difficult.

At the position level, using windowed surprise ratios across 49 matched films, we found a weak but significant effect: turning point windows show 1.4% higher surprise (Wilcoxon $p = 0.041$, 30/49 films above diagonal). It’s worth observing, but the effect size is small and larger annotated datasets would be needed to say more.

Our hard negative variant (Small-HardNeg) significantly underperformed the standard Small model (0.373 vs. 0.532 Hits@1). The hard negative sampling strategy drew negatives from nearby scenes and from the same-position between scripts, making training much harder. But since evaluation uses random negatives, the model over-specialized on distinguishing difficult cases it would rarely encounter at test time. The higher best dev loss (2.93 vs. 2.47) suggests the model never found as strong a representation as our better performing one.

Several limitations qualify our findings. We report no confidence intervals on any metrics; the context-length results at positions 75 and 100 ($n = 70$ and $n = 59$) are based on smaller subsets of longer screenplays that may not be representative. The Round 1 ablations (shuffled-order, no-

positional) were run on 149 screenplays with the Large model and were not re-run at full scale. Our findings are directionally informative but not definitive. Additionally, our corpus is entirely English-language Hollywood screenplays from IMSDB, which encodes Western narrative conventions and may not generalize to other storytelling traditions. Scene text is truncated to 2,000 characters before embedding, and MiniLM’s 256-token context window means the tail of longer scenes is not represented. Finally, we evaluate on parser-defined scenes (split on slug lines). This inflates scene counts relative to dramaturgical definitions (which we define as a sequence count) as moments spanning multiple locations would themselves contain multiple slugs.

7 Conclusion

In this project, we presented the CausalSceneTransformer, a self-supervised model that learns progressive narrative representations from screenplays using contrastive next-scene prediction. Without any labeled data, our model nearly matches a strong surface-similarity baseline on next-scene ranking while outperforming it on longer contexts and overall ranking accuracy. Linear probing further proves that narrative position and act structure emerge in the learned representations as a byproduct of the training objective, presenting the clearest evidence that our model captures something beyond surface-level scene similarity.

The most important practical finding is that data scaling, not model scaling, drove our gains. A 1.4M-parameter model easily outperformed the 13.4M-parameter model on the same data. For self-supervised narrative understanding, more stories matter more than more parameters.

Several directions for future work remain open. Larger annotated benchmarks would allow stronger evaluation of turning point correlation. Curriculum-based hard negative annealing could improve the difficult negatives use case without resulting in the training-evaluation mismatch we observed. And embedding models with longer context windows could capture full scenes rather than truncated fragments, potentially strengthening the representations our model builds on top of and lending themselves better to actually qualifying what types of narrative tracking are being integrated.

References

- [1] David Bamman, Ted Underwood, and Noah A. Smith. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 370–379, 2014.
- [2] Apoorv Agarwal, Sriramkumar Balasubramanian, Jiehan Zheng, and Sarthak Dash. Parsing screenplays for extracting social networks from movies. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature*, pages 50–58, 2014.
- [3] Philip John Gorinski and Mirella Lapata. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, 2015.
- [4] Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie plot analysis via turning point identification. 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 1707–1717, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [5] Snigdha Chaturvedi, Mohit Iyyer, and Hal Daumé III. Unsupervised learning of evolving relationships between literary characters. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [6] Marti A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.
- [7] Pinelopi Papalampidi, Frank Keller, Lea Frermann, and Mirella Lapata. Screenplay summarization using latent narrative structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1920–1933. Association for Computational Linguistics, 2020.

- [8] Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3693–3699, 2019.
- [9] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [10] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, pages 3982–3992, 2019.