

---

# Data Literacy 2025 Project Report

---

Ansel Cheung<sup>\*1</sup> Alessio Villa<sup>\*2</sup> Bartol Markovinović<sup>\*3</sup> Martín López de Ipiña<sup>\*4</sup> Niklas Abraham<sup>\*5</sup>

## Abstract

Cultural narratives encode and transmit evolving societal values, yet quantifying how meanings change over time remains methodologically challenging. This project investigates semantic evolution in cinema by analyzing how genres and thematic clusters shift within a unified semantic space across multiple decades. By representing movies as embeddings and tracking their geometric trajectories measuring velocity, acceleration, and curvature we distinguish periods of gradual semantic drift from moments of structural reorganization in cinematic history. This framework provides a quantitative foundation for understanding cultural change at scale and tests whether established linguistic laws of semantic evolution extend to film as a cultural medium.

## 1. Introduction

Cinema provides a rich archive of narrative structures that encode evolving societal values across generations. Stories serve not only to entertain but to instruct, and those narratives that align with existing social values are more likely to survive and propagate through collective memory. Recent computational work has revealed hidden cultural patterns in large narrative corpora. (?) used word embeddings to uncover systematic gender stereotypes in movie synopses, revealing the "Cinderella complex" where female characters' happiness depends asymmetrically on male characters. (?) applied topic modeling to investigate genre structure and temporal evolution, demonstrating that lexical features cap-

ture meaningful genre conventions and showing how genres shift in composition over time. These studies establish that quantitative methods can illuminate cultural phenomena at scales beyond traditional close reading, revealing patterns that operate across thousands of narratives.

However, measuring semantic change in cultural narratives over extended historical periods remains methodologically challenging. Previous approaches have examined genre structure at specific moments or through discrete topic models that capture lexical shifts but not the continuous geometric evolution of semantic categories. Can we characterize not merely that genres change, but how they change whether through gradual drift, sudden discontinuities, or cyclical patterns? Furthermore, while linguistic corpora have been analyzed for semantic drift using diachronic word embeddings, these methods require temporal alignment procedures that introduce potential artifacts when comparing meanings across decades.

We address these questions by constructing a unified semantic space from a large corpus of film plot summaries spanning multiple decades. By embedding all narratives into a single static vector space, we eliminate temporal alignment requirements while preserving fine-grained semantic relationships. Within this space, we represent genres and thematic clusters as centroids and track their trajectories over time. By computing geometric properties of these trajectories including velocity, acceleration, and curvature we can distinguish periods of continuous semantic evolution from moments of structural reorganization where genres undergo fundamental conceptual shifts. This geometric analysis reveals not just that meanings change, but the dynamics of how they change, providing quantitative measures of cultural evolution.

<sup>\*</sup>Equal contribution <sup>1</sup>Matrikelnummer 7274374, MSc Machine Learning <sup>2</sup>Matrikelnummer 7306912, MSc Computer Science <sup>3</sup>Matrikelnummer 7324790, MSc Machine Learning <sup>4</sup>Matrikelnummer 7293076, MSc Machine Learning <sup>5</sup>Matrikelnummer 7307188, MSc Machine Learning. Correspondence to: Initials1 <ansel-heng-yu.cheung@uni-tuebingen.de>, Initials2 <alessio.villa@student.uni-tuebingen.de>, Initials3 <bartol.markovinovic@student.uni-tuebingen.de>, Initials4 <martin.lopez-de-ipina-munoz@student.uni-tuebingen.de>, Initials5 <niklas-sebastian.abraham@student.uni-tuebingen.de>.

*Project report for the "Data Literacy" course at the University of Tübingen, Winter 2025/26 (Module ML4201). Style template based on the ICML style files 2025. Copyright 2025 by the author(s).*

---

## 2. Data and Methods

### 2.1. Data Collection

The construction of our movie corpus required a multi-stage data collection pipeline that systematically integrated information from three complementary sources: Wikidata, The Movie Database (TMDb), and Wikipedia. This approach leverages the structured metadata capabilities of knowledge bases while obtaining rich textual descriptions necessary for semantic analysis.

The initial data collection phase queried Wikidata, a collaboratively edited multilingual knowledge graph maintained by the Wikimedia Foundation. Wikidata serves as an ideal entry point for systematic movie data collection due to its comprehensive coverage of cultural artifacts and its structured representation of temporal, categorical, and relational metadata. For each year in our study period, we retrieved movies satisfying specific criteria, including release year, film classification, and the availability of linked Wikipedia articles in English. This step yielded essential metadata fields including unique Wikidata identifiers, film titles, release years, and crucially, sitelinks to corresponding Wikipedia articles.

The second enrichment stage leveraged The Movie Database (TMDb), a community-maintained database that provides quantitative popularity metrics and user engagement statistics. To bridge between the Wikidata entities and TMDb records, we exploited the external identifier mappings maintained within Wikidata. Specifically, each Wikidata identifier was used to query the TMDb API, retrieving popularity scores, vote averages, and vote counts for each film. These metrics serve as proxy measurements for audience engagement and cultural impact, enabling downstream filtering and weighting of films based on their relative prominence within the cinematic landscape. This cross-database linkage was essential as TMDb contains richer engagement data than Wikidata, while Wikidata provides superior structured coverage of historical films.

The third and most data-intensive stage involved retrieving full-text plot descriptions from Wikipedia articles. Using the English Wikipedia sitelinks obtained from Wikidata, we systematically accessed each film’s Wikipedia page and extracted the plot summary section. Wikipedia’s editorial policies ensure that plot descriptions maintain a relatively consistent level of detail and neutral tone across articles, providing standardized narrative summaries suitable for comparative semantic analysis. The extraction process employed the Wikipedia API to retrieve article content, followed by section identification and text normalization procedures. This step transformed the structured metadata from the previous stages into the rich textual data required for embedding-based analysis.

All data sources are openly accessible under permissive li-

censes. Wikidata (?) provides structured data under the CC0 1.0 Universal license (public domain dedication), enabling unrestricted use without attribution requirements. Wikipedia (?) content is licensed under CC BY-SA 4.0, requiring attribution and share-alike distribution of derivative works. TMDb (?) data is available under CC BY-NC 4.0 for non-commercial research purposes with appropriate attribution. This licensing framework ensures the reproducibility and legal compliance of our data collection pipeline.

The resulting dataset exhibits high coverage across decades, with successful retrieval of plot descriptions for over 80% of films in most decades, as shown in Table ???. The sequential pipeline design ensures data consistency through unique identifiers while maximizing coverage by combining the complementary strengths of each data source.

*Table 1.* Per-decade feature coverage (%) of key metadata fields in the movie dataset.

Decade	Actors+Director	Genre	Plot	Vote Count
1950s	86.56	63.67	83.03	90.19
1960s	83.82	61.01	77.31	85.26
1970s	86.55	62.47	79.57	86.58
1980s	84.68	59.41	79.80	85.41
1990s	82.22	58.33	82.00	84.49
2000s	77.34	60.51	84.19	83.73
2010s	70.25	60.55	84.94	85.60
2020s	70.66	64.85	77.59	89.63
Average	80.26	61.35	81.05	86.36

After the data was collected in a tabular format, the textual plot descriptions required transformation into vector representations suitable for downstream analysis. The plot descriptions extracted from Wikipedia pages exhibit substantial variability in length, ranging from 10 to 20,479 characters, corresponding to approximately 6 to 5,296 tokens. All plot descriptions in our corpus are in English, which simplifies the embedding process by eliminating cross-lingual considerations.

The selection of an appropriate embedding model was guided by the Massive Text Embedding Benchmark (MTEB) leaderboard results<sup>1</sup>, which provides comprehensive evaluations of embedding models across diverse retrieval and semantic similarity tasks. Based on these benchmarks, we selected the BGE-M3 (Beijing Academy of Artificial Intelligence Multilingual, Multifunctional, Multi-granularity) model (?), an open-source model developed by the Beijing Academy of Artificial Intelligence. The BGE-M3 model achieved competitive performance (28th place on the MTEB leaderboard) while maintaining a relatively compact archi-

<sup>1</sup><https://huggingface.co/spaces/mteb/leaderboard>

---

ture with 0.5 billion parameters. Critically, the model supports a context length of 8,192 tokens, which enables embedding entire movie plot descriptions into a single vector representation without requiring chunking for the majority of documents in our corpus.

The BGE-M3 model offers three distinct embedding modes, each suited for different retrieval and analysis tasks. The *dense vector* output corresponds to the [CLS] token representation from the final transformer layer, producing a 1024-dimensional vector that serves as a global document representation (?). The *sparse vector* mode generates token-level weights with extremely high dimensionality (250,002 dimensions), where each token is represented by a single weight, enabling fine-grained lexical matching. The *multi-vector* output provides all hidden states from the model, yielding 1024-dimensional vectors for each token in the input sequence, thus preserving token-level semantic information.

A critical methodological decision in our analysis was the use of a single static embedding model across all time periods, rather than training separate temporal models or employing fine-tuning procedures. This approach offers several advantages while relying on a key assumption about our data. By maintaining a unified latent space, we avoid the need for post-hoc alignment transformations between differing embedding spaces a procedure that would introduce unnecessary complexity and potential alignment artifacts. While films from earlier decades may employ substantially different linguistic conventions than contemporary cinema, the plot descriptions in our corpus are not historical documents from those eras. Wikipedia’s collaborative editing model ensures that article content, including plot summaries, undergoes continuous revision and updating by contemporary editors. Consequently, the language used in these descriptions reflects modern English usage, with the all of summaries having been written or substantially revised within approximately the past two decades (Wikipedia, was founded in 2001). We therefore assume that semantic drift in the descriptive language itself is minimal across our corpus, even as the described narratives span multiple decades of cultural history. This baseline assumption that plot descriptions maintain consistent linguistic conventions due to Wikipedia’s living document nature underlies all subsequent embedding analyses in this work and enables direct comparison of semantic representations across time periods without requiring temporal alignment procedures.

A fundamental challenge in embedding variable-length documents arises from the inherent limitations of transformer-based models. Our corpus contains documents ranging from a few sentences to several thousand words, while transformer models exhibit fixed token limits and demonstrate biases in their learned representations. Specifically, the

[CLS] token, which is commonly used as a document-level representation, has been shown to exhibit length-dependent biases and structural preferences (?). Therefore, selecting an appropriate pooling and chunking strategy is essential to minimize length bias while preserving semantic fidelity across documents of varying lengths.

While the [CLS] token approach appears attractive due to its simplicity and the model’s pre-training objective of learning document-level representations, empirical evidence demonstrates significant limitations for long documents. (?) showed that attention entropy at the [CLS] token drops sharply as sequence length increases, with the token focusing predominantly on the first approximately 128 tokens. Similarly, (?) reported that learned summary tokens underperform mean pooling for long documents, with decoder attention mechanisms favoring earlier positions in the sequence. Given that over 75% of the plot descriptions in our corpus exceed 512 tokens, relying solely on the [CLS] token would result in substantial information loss for the majority of documents.

We considered four primary approaches for generating document embeddings, visualised in Figure ??:

**Mean Pooling (Global Average):** This classical approach computes the mean of all token embeddings to produce a single document representation. While this method exhibits low variance, it introduces high bias by treating all tokens equally, potentially diluting important semantic information.

**Chunk-then-Embed (Early Chunking):** This strategy splits long documents into smaller chunks before embedding, processes each chunk separately, and then aggregates the resulting embeddings. This approach, exemplified by hierarchical attention networks (?), can reduce bias but introduces higher variance due to the loss of cross-chunk contextual information.

**Embed-then-Chunk (Late Chunking):** This method, recently proposed by (?), embeds the full text in a single forward pass through the transformer model, then pools token embeddings over fixed-size windows after contextualization. This approach maintains full contextual information while providing controlled variance and minimal bias.

**CLS Token Extraction:** The simplest approach utilizes only the [CLS] token from the final layer as the document representation, relying on the model’s learned summarization capabilities.

To determine the optimal chunking and pooling strategy for our specific use case, we conducted comparative tests across multiple approaches.

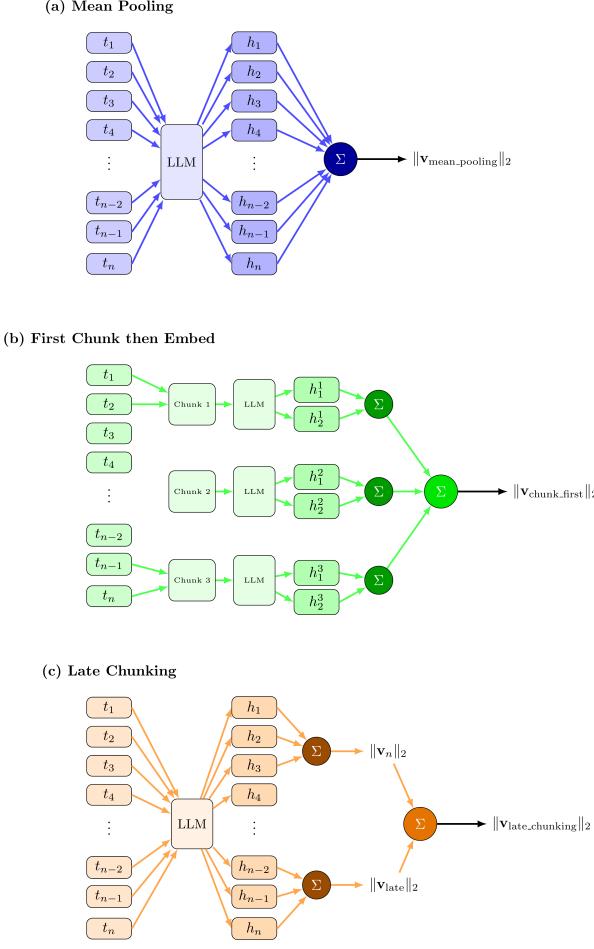


Figure 1. Comparison of different chunking methods: (a) Mean Pooling, (b) First Chunk then Embed, and (c) Late Chunking.

## 2.2. Methodology

Once movie plots are embedded into a unified semantic space, quantitative analysis of their geometric relationships becomes possible through distance metrics. The cosine distance between embeddings provides a natural measure of semantic dissimilarity, enabling the construction of cumulative distribution functions over pairwise distances within defined subsets of the corpus. Such distributions encode structural properties of the embedding space and reveal whether semantic relationships exhibit systematic patterns across temporal periods or thematic categories.

To rigorously compare distance distributions across different subsets of movies, for instance films from different decades or belonging to distinct genres, we employ the Kolmogorov-Smirnov test, a non-parametric statistical method for assessing whether two empirical distributions arise from the same underlying continuous distribution (?). The two-sample KS test compares the empirical cumulative distribution functions (ECDFs) of two samples by computing the maximum

vertical distance between them.

Formally, given two samples  $X_1, \dots, X_n$  and  $Y_1, \dots, Y_m$ , their empirical cumulative distribution functions are defined as:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n 1_{X_i \leq x}, \quad G_m(y) = \frac{1}{m} \sum_{j=1}^m 1_{Y_j \leq y} \quad (1)$$

where  $1$  denotes the indicator function. The KS test statistic is defined as the supremum of absolute differences between these ECDFs:

$$D_{n,m} = \sup_x |F_n(x) - G_m(x)| \quad (2)$$

Under the null hypothesis that both samples are drawn from the same continuous distribution, the distribution of  $D_{n,m}$  is known and can be used to compute p-values for hypothesis testing. The test is particularly suited for our application because it makes no assumptions about the underlying distributional form, is sensitive to differences in both location and shape, and operates directly on the distance measurements without requiring binning or parametric modeling.

A key design choice in applying this framework is the selection of a reference point from which distances are computed. One natural approach is to use the mean vector of a baseline subset of movies as a reference embedding, then compute the distribution of distances from this reference point to all movies in the corpus. This baseline distribution can then serve as a comparison standard against which distance distributions from other subsets are evaluated. For instance, the mean embedding of movies from a particular decade or genre can be computed, and the resulting distance distribution compared against the baseline using the KS test to assess whether the subset exhibits systematically different spatial organization relative to the reference.

In the context of temporal semantic analysis, this framework enables systematic comparison of distance distributions across decades. If the semantic structure of cinema remains stable over time, distance distributions should remain statistically similar. Conversely, significant differences in these distributions, as detected by the KS test, would indicate structural reorganization of the semantic space, suggesting periods where narrative conventions undergo fundamental shifts. By applying this analysis to distances measured from fixed reference points in the embedding space, we can quantify how the density and dispersion of semantic representations evolve temporally, providing a complementary perspective to trajectory-based analyses of genre evolution.

## 2.3. Genre analysis

As movie genres provide a meaningful taxonomy with potential temporal evolution patterns, we examine semantic

drift across different time periods. To this end, embeddings are first grouped by genre  $g$  into discrete time periods  $\tau$ , forming the set  $\mathcal{M}_g^{(\tau)}$  of plot embeddings. For each group, two alternative representative embeddings are computed: the **centroid** (arithmetic mean)  $\bar{\mathbf{e}}_g^{(\tau)}$  and the **medoid** (cosine distance minimizer embedding)  $\tilde{\mathbf{e}}_g^{(\tau)}$ .

With an arbitrary number of years  $\Delta t$  per group, the period index  $\tau$  is calculated by flooring the movie year to the nearest multiple of  $\Delta t$ :

$$\tau = \left\lfloor \frac{\text{year}}{\Delta t} \right\rfloor \cdot \Delta t \quad (3)$$

We computed the following metrics to analyse the drift dynamics across the groups:

**Genre drift and acceleration:** drift (Equation ??) measures displacement between representative embeddings of consecutive periods, capturing how a genre’s semantic center evolves over time. Acceleration quantifies the change in drift between consecutive periods.

$$\mathbf{d}_g^{(\tau)} = \bar{\mathbf{e}}_g^{(\tau+\Delta t)} - \bar{\mathbf{e}}_g^{(\tau)} \quad (4)$$

**Inter-genre distance:** determines cosine distance between representatives of each pair of genres for each year, enabling pairwise comparison between specific genres.

Due to group size differences between time periods, two alternative normalization approaches have been employed: (1) downsampling, ensuring equal sampling error across groups, and (2) z-score normalization, which accounts for the standard error of the difference between group means:

$$\hat{v}_g^{(\tau)} = \frac{v_g^{(\tau)}}{\sigma_{\text{pooled}} \cdot \sqrt{\frac{1}{n_g^{(\tau)}} + \frac{1}{n_g^{(\tau+\Delta t)}}}} \quad (5)$$

where  $\sigma_{\text{pooled}}$  is the pooled within-group standard deviation of cosine distances, and  $n_g^{(\tau)}$  is the number of movies in genre  $g$  at time  $\tau$ .

None of the genre-based analyses yielded statistically significant results, suggesting that genres may be too broad as analytical categories and any underlying patterns are likely obscured by noise. Figure ?? illustrates temporal drift for the three most popular genres, computed over 5-year periods. Each period was downsampled to 145 movies, with 95% confidence intervals estimated via bootstrapping (1,000 samples).

**Projection along temporal axis** This idea is to select an arbitrary vector that is in the embedding space and project the movie embeddings onto this vector. Based on the choice of this vector, we will be able to see how much of each movie embedding lies on the vector, and we can then do temporal

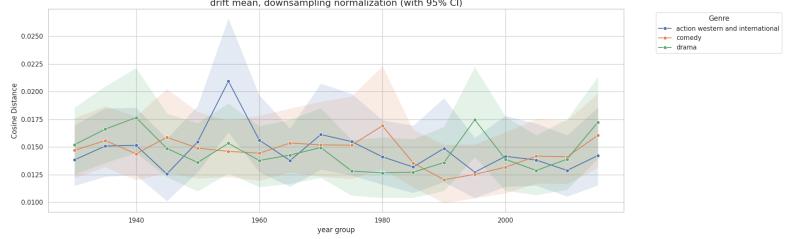
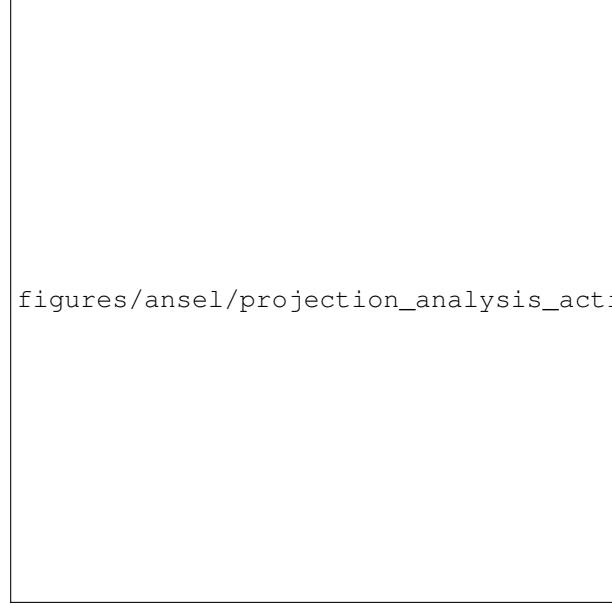


Figure 2. Genre drift from 1930 to 2025 of three most popular genres over 5-year periods

analysis to see how this metric evolves over time. Since our embeddings are already normalized to magnitude 1, cosine distance is proportional to L2 norm, which measures euclidean distance between embeddings. Note that for this Projection Analysis, all experiments were bootstrapped for 500 times, 1000 samples.



figures/ansel/projection\_analysis\_action\_1930\_2024

Figure 3. Projection onto action vector over years

First vector we chose was  $\text{mean}(\text{emb}_{\text{action} 2024}) - \text{mean}(\text{emb}_{\text{action} 1930})$ . From Figure ??, we can see that movie plots are increasingly becoming similar to action movie plots. If we look at Figure ?? we also see the same trend for Romance. This created suspicion and we plotted the same thing but the overall centroid shift from 1930 to 2024 and also saw the same plots (Figure ??). This could mean that how movie plots evolve over time outweighs the shift in genres.

In order to then explore the shift in genres over time, we plotted the Cosine Distance evolution of mean action embedding per year from the overall centroid. From Figure ?? we

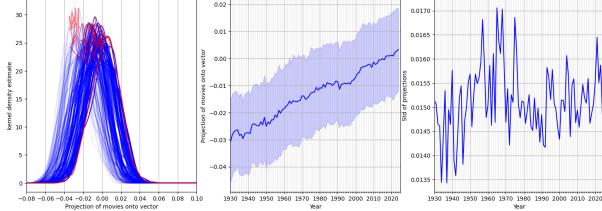


Figure 4. Projection onto Romance vector over years

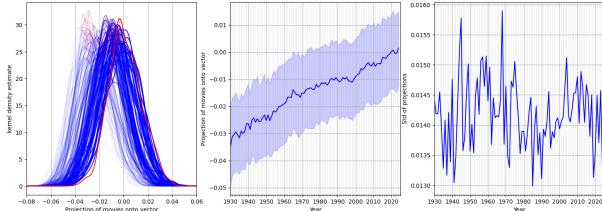


Figure 5. Projection onto Mean vector over years

observe that the action movies are getting more and more similar to the average embedding vector. The Romance cosine distance to the same mean vector shows a different story (Figure ??).

Figure ?? We expanded this to our new genres and this revealed that many large genres were converging towards the mean embedding. There are some outliers such as "Anime" which only appeared after 1980. "Film Noir" and "Adventure and Fantasy" did not follow the trend of converging towards the mean embedding. (do I end here?)

**Spread analysis** Another way to see if movies are converging over the years or spreading out is to measure the spread per year. (<https://arxiv.org/pdf/1810.08693>) Frobenius norm measures the total variance of each year's difference in movie embeddings to its yearly mean embedding. The frobenius norm (<https://arxiv.org/pdf/1501.01571> page 84 ) is the sum of squared singular values in which we are only measuring noise and not the signal. In order to see the signal shift, we use the spectral norm to find the maximum singular value of the difference in movie embeddings and their respective yearly mean embeddings.

We observe from Figure ?? that movies are getting more and dissimilar. Paired together with the mean L2 distance from each movie to the yearly mean embedding, we can see that the average distance from each movie to its yearly mean remains the same, but the spectral norm triples in size (Figure ??), signalling that there is some sort of shape shift or stretching of the embedding cloud. There is a sharp drop in 2020, which might be attributed to movie production during covid period reducing the number of movies being produced and hence reducing the chance for outliers.

The next sensible step to take is to analyze the explained

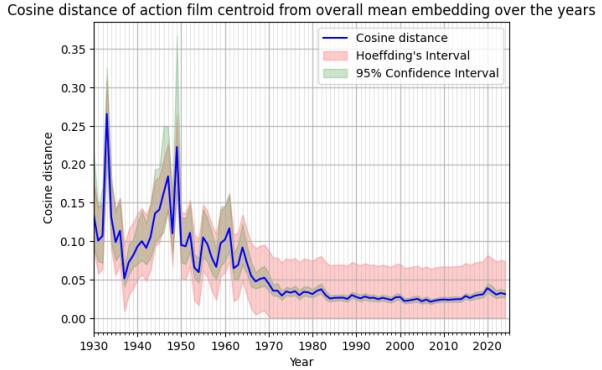


Figure 6. Action cosine distance to mean (all) embedding vector

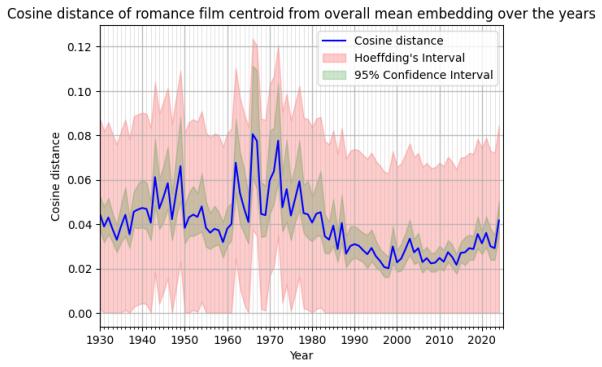


Figure 7. Romance cosine distance to mean (all) embedding vector

variance of the first principal component. This can simply be calculated by the squared spectral norm divided by the squared frobenius norm. From Figure ??, the explained variance increases from 3% to 4% and sharply rises to 4.75% after 2020. An explained variance of 4% is significant in a dataset with dimension of 1024. If all dimensions were random noise, each PC would explain  $1/1024 \approx 0.098\%$ , so 4% is 40 times higher than random. This means that there is a direction which is polarizing the movie industry.

This is actually one large circular path leading back to PCA. Now we can interpret the evolution of movies to see which movies are the most polarized by that particular year's principal component. Hence we project every movie's embeddings onto its year's PC. From here we select an arbitrary number of years to analyze its top and bottom 5 movies.

(PCA1 table here)

(PCA US Movies)

(PCA German Movies)

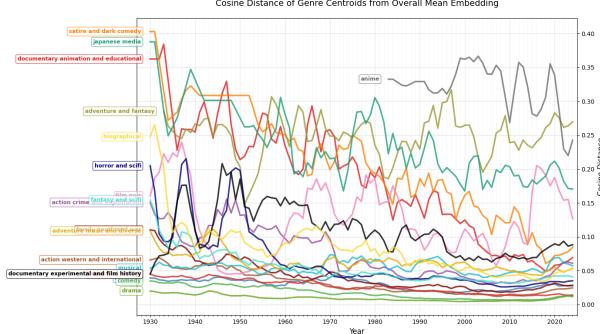


Figure 8. All genre's cosine distance to mean (all) embedding vector

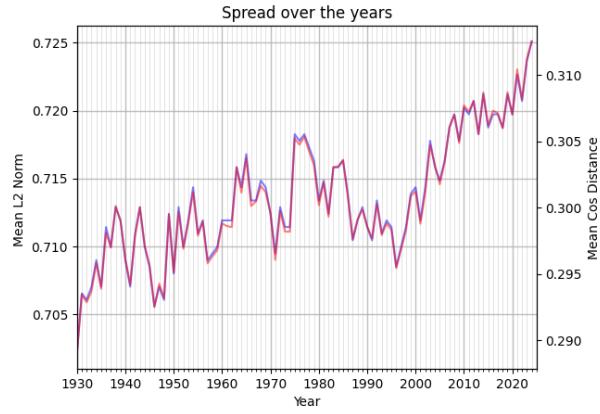


Figure 9. Mean L2 Norm vs Cosine Distance against yearly centroid

### 3. Results

#### 3.1. Chunking Method Comparison

We evaluated ten distinct embedding strategies across 5,000 movie plot descriptions, comprising four fundamental approaches with various parameterizations: Mean Pooling, CLS Token extraction, Chunk-First-Embed with three configurations (512/256, 1024/512, 2048/1024 tokens for chunk size and stride), and Late Chunking with six configurations. The evaluation employed multiple metrics to assess embedding quality, including length bias, isotropy, genre classification performance, and class separation characteristics. Table ?? presents key performance metrics across all methods.

**Length Bias and Correlation:** The length-normalization correlation metric reveals substantial variation across methods. MeanPooling and LateChunking variants exhibit positive correlations ranging from 0.62 to 0.82, with LateChunking\_512\_256 achieving the highest value (0.82). In contrast, CLSToken demonstrates near-zero correlation (0.0035), while ChunkFirstEmbed methods show negative correlations ranging from -0.03 to -0.37, with ChunkFirstEm-

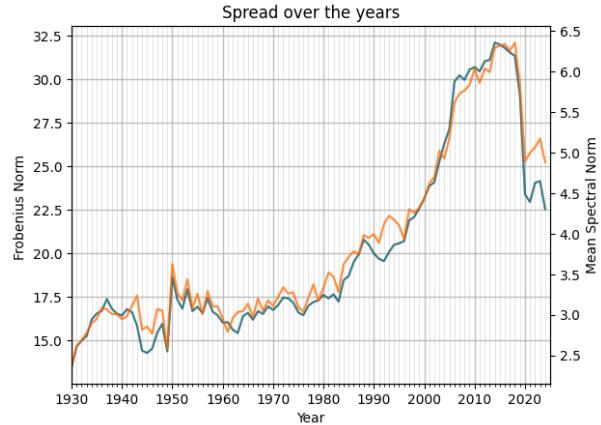


Figure 10. Spectral and Frobenius Norm of yearly movies against yearly centroid

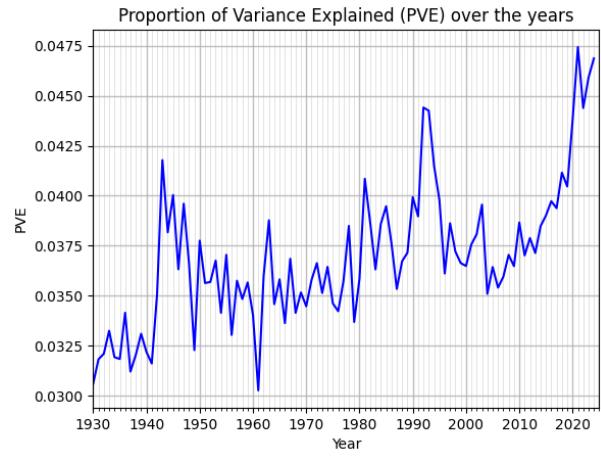


Figure 11. PC1 Explained Variance of yearly movies against yearly centroid

bed\_512\_256 exhibiting the strongest negative correlation (-0.37).

**Isotropy Measurements:** Isotropy, measured by the proportion of variance captured in the first principal component, shows a clear distinction between method families. Mean-Pooling and LateChunking variants demonstrate substantially lower isotropy (first PC: 11.27-11.92%), indicating more uniformly distributed embeddings across dimensions. CLSToken and ChunkFirstEmbed methods exhibit significantly higher isotropy (first PC: 3.32-3.47%), suggesting greater concentration of variance along principal axes. After removing the top two principal components, isotropy values converge to a narrower range (2.27-3.09%) across all methods.

**Genre Classification Performance:** Genre classification accuracy, evaluated using a logistic regression classifier,

**Table 2.** Comparative performance of chunking methods across key evaluation metrics. Length-Norm Corr measures correlation between document length and embedding norm. Isotropy (1st PC) indicates variance concentration in the first principal component (lower is better). Genre Acc and F1 report classification performance. Silhouette measures cluster cohesion. Sep. Ratio quantifies intra-class to inter-class similarity ratio.

Method	Length-Norm Corr	Isotropy (1st PC %)	Genre Acc	Genre F1	Silhouette	Sep. Ratio
Mean Pooling	0.629	11.36	0.326	0.180	-0.036	0.943
CLS Token	0.004	3.32	0.341	0.194	-0.016	0.958
Chunk-First 512/256	-0.366	3.47	0.349	0.198	-0.020	0.951
Chunk-First 1024/512	-0.275	3.37	0.348	0.197	-0.017	0.959
Chunk-First 2048/1024	-0.031	3.33	0.341	0.194	-0.016	0.961
Late Chunk 512/256	0.822	11.92	0.327	0.180	-0.037	0.949
Late Chunk 1024/512	0.726	11.53	0.326	0.181	-0.037	0.948
Late Chunk 2048/1024	0.656	11.36	0.326	0.180	-0.036	0.958
Late Chunk 2048/512	0.656	11.36	0.326	0.180	-0.036	0.958
Late Chunk 512/0	0.821	11.40	0.328	0.183	-0.037	0.948

demonstrates modest variation across methods. Accuracy ranges from 0.326 to 0.349, with `ChunkFirstEmbed_1024.512` achieving the highest accuracy (0.349) and `LateChunking_512.256` the lowest (0.327). Macro-averaged F1 scores similarly cluster between 0.179 and 0.198, with `ChunkFirstEmbed` methods showing slight superiority (F1: 0.193-0.198) over `LateChunking` and pooling-based approaches (F1: 0.178-0.183). Silhouette scores, measuring cluster cohesion, remain consistently negative across all methods (ranging from -0.037 to -0.016), indicating substantial overlap in genre representations.

**Class Separation Metrics:** Intra-class cosine similarity (mean similarity within genre groups) exhibits narrow variation from 0.500 to 0.533 across methods. Inter-class similarity (mean similarity between different genres) similarly ranges from 0.481 to 0.509. The resulting separation ratios, computed as the ratio of intra-class to inter-class similarity, range from 0.943 to 0.962, with `ChunkFirstEmbed_2048.1024` achieving the highest separation (0.961). Separation gaps, defined as the difference between intra- and inter-class similarity, range from 0.019 to 0.030, with `MeanPooling` showing the largest gap (0.030) and `ChunkFirstEmbed_2048.1024` the smallest (0.019).

**Cosine Similarity Distributions:** The distribution of pairwise cosine similarities across methods reveals consistent central tendencies with mean similarities clustering between 0.485 and 0.505. Standard deviations range from 0.061 to 0.126, with `MeanPooling` and `LateChunking` methods exhibiting higher variance (std: 0.123-0.126) compared to `CLSToken` and `ChunkFirstEmbed` approaches (std: 0.061-0.067). Minimum observed similarities span from 0.054 to 0.261, while maximum similarities range from 0.688 to 0.836, indicating that `MeanPooling` and `LateChunking` methods produce wider similarity distributions.

**Parameter Sensitivity in Late Chunking:** Within the Late-

Chunking family, window size and stride parameters demonstrate measurable effects on performance characteristics. Configurations with stride 0 (non-overlapping windows) show slightly improved silhouette scores (from -0.037 to -0.037 compared to overlapping configurations). Larger window sizes (2048 tokens) consistently yield higher separation ratios (0.958) compared to smaller windows (512 tokens: 0.943-0.948).

## 4. Discussion & Conclusion

Use this section to briefly summarize the entire text. Highlight limitations and problems, but also make clear statements where they are possible and supported by the analysis.

---

## Contribution Statement

Explain here, in one sentence per person, what each group member contributed. For example, you could write: Max Mustermann collected and prepared data. Gabi Musterfrau and John Doe performed the data analysis. Jane Doe produced visualizations. All authors will jointly wrote the text of the report. Note that you, as a group, a collectively responsible for the report. Your contributions should be roughly equal in amount and difficulty.

## Notes

Your entire report has a **hard page limit of 4 pages** excluding references and the contribution statement. (I.e. any pages beyond page 4 must only contain the contribution statement and references). Appendices are *not* possible. But you can put additional material, like interactive visualizations or videos, on a githunb repo (use [links](#) in your pdf to refer to them). Each report has to contain **at least three plots or visualizations**, and **cite at least two references**. More details about how to prepare the report, inclucing how to produce plots, cite correctly, and how to ideally structure your github repo, will be discussed in the lecture, where a rubric for the evaluation will also be provided.

## References

- Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., and Liu, Z. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024. URL <https://arxiv.org/abs/2402.03216>.
- Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL <http://arxiv.org/abs/1810.04805>.
- Gong, L., He, D., Li, Z., Qin, T., Wang, L., and Liu, T. Efficient training of BERT by progressively stacking. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2337–2346. PMLR, 09–15 Jun 2019. URL <https://proceedings.mlr.press/v97/gong19a.html>.
- Günther, M., Mohr, I., Williams, D. J., Wang, B., and Xiao, H. Late chunking: Contextual chunk embeddings using long-context embedding models, 2025. URL <https://arxiv.org/abs/2409.04701>.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL <https://arxiv.org/abs/1910.10683>.
- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, San Diego, California, jun 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1174. URL <https://aclanthology.org/N16-1174/>.