Data Literacy 2025 Project Report

Ansel Cheung [*1] **Alessio Villa** [*2] **Bartol Markovinović** [*3] **Martín López de Ipiña Muñoz** [*4] **Niklas Abraham** [*5]

Abstract

This project addresses the fundamental question of how cultural meaning evolves over time by quantitatively modeling seventy-five years of cinematic history through 200,000 film synopses embedded in a single static semantic space using the BGE-M3 model. Temporal change is measured by tracking the movement of genre centroids within this space—analyzing their velocity, acceleration, and curvature to distinguish continuous evolution from structural paradigm shifts. The framework provides a reproducible and data-driven foundation for cultural analytics, testing whether established linguistic laws of semantic drift extend to the domain of cinema.

1. Introduction

Motivate the problem, situation or topic you decided to work on. Describe why it matters (is it of societal, economic, scientific value?). Outline the rest of the paper (use references, e.g. to Section 2: What kind of data you are working with, how you analyse it, and what kind of conclusion you reached. The point of the introduction is to make the reader want to read the rest of the paper.)

2. Data and Methods

Data collection process, bla bla bla, the pipline explained, etc.

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.

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

The selection of an appropriate embedding model was guided by the Massive Text Embedding Benchmark (MTEB) leaderboard results[1], 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 (Chen et al., 2024), 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 architecture 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

[*]Equal contribution [1]Matrikelnummer 7274374, MSc Machine Learning [2]Matrikelnummer 7306912, MSc Computer Science [3]Matrikelnummer 7324790, MSc Machine Learning [4]Matrikelnummer 7293076, MSc Machine Learning [5]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.

[1]https://huggingface.co/spaces/mteb/leaderboard

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 (Devlin et al., 2018). 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 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 (Devlin et al., 2018). 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. (Gong et al., 2019) 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, (Raffel et al., 2023) 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 1:

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 (Yang et al., 2016), can reduce bias but introduces higher variance due to the loss of cross-chunk contextual information.

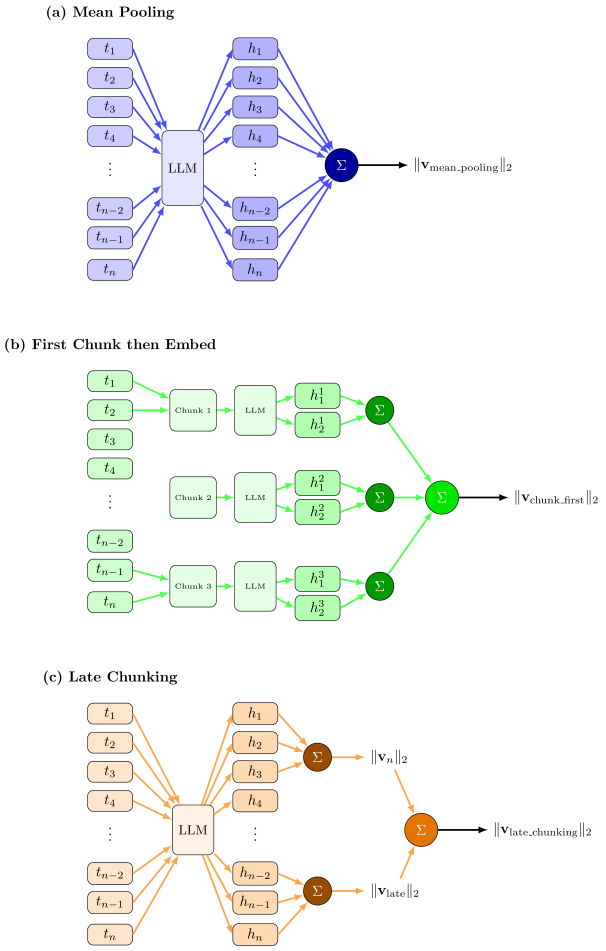

Figure 1. Comparison of different chunking methods: (a) Mean Pooling, (b) First Chunk then Embed, and (c) Late Chunking.

Embed-then-Chunk (Late Chunking): This method, recently proposed by (Günther et al., 2025), 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.

3. Results

In this section outline your results. At this point, you are just stating the outcome of your analysis. You can highlight

important aspects ("we observe a significantly higher value of x over y"), but leave interpretation and opinion to the next section. This section absoultely *must* include at least two figures.

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/.