Temporal Pattern Exploration in News Videos

Faculty of Electrical Engineering and Computer Science

Institute of Data Science

Leibniz University Hannover

Master Thesis

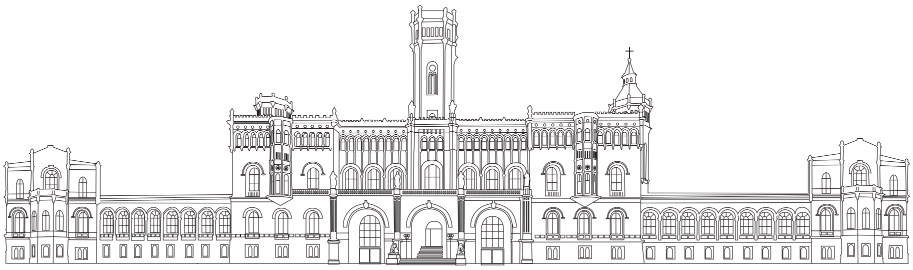
submitted for the degree of

Master of Science (M. Sc.)

by

Lea Rebecca Reinhart

Matriculation Number : 10054743



First Examiner: Prof. Dr. Ralph Ewerth

Second Examiner: Prof. Dr. Sören Auer

Supervised By: Dr. Eric Müller-Budack

April 7, 2025

Erklärung der Selbständigkeit

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt wurden, alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind, und die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt habe.

Hannover, April 7, 2025

Lea Rebecca Reinhart

Abstract

In recent years, video has become a dominant format in the news media landscape. Understanding how structural and stylistic choices influence audience perception is essential and detecting such choices at scale is a necessary first step. This thesis investigates temporal patterns in news videos, specifically film editing patterns (FEPs) and narrative strategies (NSs), to assess their prevalence, editorial function, and detectability using machine learning techniques. Drawing on a curated, multimodal dataset of 128 news stories from five ideologically diverse German broadcasters, the study addresses three core research questions focused on pattern exploration, feature analysis, and model performance. The proposed methodology combines interpretable features from visual, audio, and multimodal sources with ensemble classifiers (Random Forest, Gradient Boosting), and compares this approach to a zero-shot vision-language model. Exploratory analysis reveals consistent stylistic differences between outlets, frequent co-occurrence between certain FEPs and NSs, and a higher density of patterns in privately owned channels. Empirical findings show that visual features are particularly effective for detecting FEPs, while multimodal features boost performance for NS detection. In general, targeted feature selection improves generalization and recall. Per-class classifiers outperform multi-label models, especially in terms of recall, whereas current vision-language models underperform in structured pattern recognition tasks. This thesis presents the first systematic, automated approach to detecting film editing patterns and narrative strategies in news videos. It offers practical tools and theoretical insights for computational media analysis, while critically highlighting key challenges such as annotation subjectivity and cross-domain generalization. Ultimately, the work lays a foundation for scalable, interpretable systems that support transparency and accountability in the analysis of broadcast journalism.

Contents

List of Tables

List of Figures

1 Introduction

In our interconnected world, individuals around the globe rely on news media to stay informed about local and international events [82]. However, news is more than factual information about events – it significantly shapes our understanding of the world [53]. This impact is particularly vital in democratic societies, where media is often deemed the unofficial fourth pillar of democracy because public beliefs can directly influence voting behavior [8, 28, 32]. Given this influence, news media has long been a political battleground [76]. It can be challenging for consumers to form balanced opinions in this battleground state of news [81]. One issue is the proliferation of fake news, which can continue influencing beliefs even after being debunked [3, 32, 66]. Furthermore, regardless of the truthfulness of the content, news videos can employ film editing patterns and narrative strategies. These refer to structural and stylistic conventions—borrowed from cinematic storytelling—that include shot composition, sequencing, emotional framing, or speaker emphasis. Such techniques can influence viewers' perceptions of content. While their usage can facilitate engagement and learning [29, 98], they can also influence the persuasiveness of messages [17], making it harder to engage with news content objectively.

1.1 Limitation of Related Work

Despite efforts to combat misinformation in text-based media through approaches such as fact-checking [48, 92, 96] and narrative analysis [52], video content has received comparatively limited attention [32]. With video emerging as a dominant format for news consumption [81], the need for research on the analysis of news videos is evident. Current analysis methods often focus on textual transcripts of news videos, disregarding the multimodal connections inherent to video content [32]. This is problematic since multimodal temporal patterns like filmic editing patterns or narrative strategies can not occur in text-based media. Existing works exploring such patterns in news videos are scarce. Bateman and Tseng [13] identified significant differences in narrative pattern usage between state-owned and private media; however, their analysis relied on expert annotations. Given the scale of news content - exacerbated by AI-generated media - reliance on human annotators is impractical. Automated approaches are critical to addressing modern news challenges at scale [120]. Historically, automated analysis of foundational temporal structures in news videos has been conducted, for example, via hierarchical decision trees and hidden Markov models [63, 90]. Contemporary research centers on more complex domains. Cheema et al. [21] utilized random forests, XGBoost, and vision-language models to classify speaker roles and situational contexts in news videos [21]. However, works in the automated analysis of temporal patterns in news videos remain sparse. Specifically, to our knowledge, there has been no published work that systematically explores filmic editing patterns and narrative strategies in news videos using machine learning techniques.

1.2 Contribution

This gap leads us to the following research questions:

- **RQ1 – Pattern Exploration:** How are film editing patterns and narrative strategies distributed across news outlets, and how do they co-occur or relate to each other?
- **RQ2 – Feature Analysis:** Which types of features—visual, audio, linguistic, or multimodal—are most informative for detecting film editing patterns and narrative strategies, and how does feature effectiveness vary by task?
- **RQ3 – Model Performance and Generalization:** Which machine learning models achieve the best performance in detecting temporal patterns in news videos, and how well do they generalize across editorial domains?

This thesis aims to address these questions and is, to our knowledge, the first systematic attempt to detect and analyze filmic editing patterns (FEPs) and narrative strategies (NSs) in news videos using automated, data-driven approaches. Bridging methods from computer science and media studies, the study processes and analyzes a curated multimodal dataset of 128 news stories from five ideologically and stylistically diverse German news outlets. Aligned with the research questions, the contributions of this work are threefold:

1. Empirical analysis of temporal patterns in news videos: Through exploratory data analysis, the thesis investigates how FEPs and NSs are distributed across public and private broadcasters, how they differ in frequency, duration, and co-occurrence, and how they are related to a variety of multimodal features. Some patterns (e.g., shot-reverse-shot, emotionalization) appear consistently across outlets, while others are highly outlet-specific. Overall, private broadcasters tend to use more patterns per story, consistent with findings by Bateman and Tseng [13].

2. Insights into feature-target relations for pattern detection: The thesis evaluates a comprehensive set of interpretable features from different modalities extracted from raw video data. It finds that visual features are essential for detecting FEPs, while multimodal and handcrafted feature sets perform best for NSs. It also explores expected and unexpected correlations between features and target patterns—for example, the expected link between shot similarity (with a two-shot gap) and alternating-shot patterns, and the surprising absence of correlations between emotionalization and detected emotional expressions.

3. Benchmarking automated models for detection and generalization: Using Random Forest and Gradient Boosting Classifiers, the thesis compares multi-class and per-class prediction strategies, evaluating their performance on both k-fold and cross-domain setups. Per-class models consistently outperform multi-class ones in recall and F-scores. Visual language models (VLMs), used as zero-shot baselines, perform poorly, highlighting the limitations of current general-purpose models for structured multimodal detection tasks.

Together, these contributions lay the groundwork for tools that can assist researchers, educators, and journalists in analyzing how news videos are structured and how they may influence audiences. The findings also help pave the way for future development of scalable datasets and interpretable models tailored to complex, real-world media analysis tasks.

1.3 Structure of the Thesis

The thesis is organized as follows: Chapter 2 reviews related work, contextualizing the study within the domains of multimodal analysis, temporal pattern detection, and their applications in news media. Chapter 3 outlines the theoretical background, including definitions and frameworks for narrative strategies and film editing patterns. Chapter 4 details the methodology, including task definitions, feature extraction, and machine learning model design. Chapter 5 presents the experimental setup and results, followed by discussion, limitations, and future work. Finally, Chapter 6 concludes the thesis by summarizing the findings and discussing their broader relevance.

2 Related Work

The computational analysis of temporal patterns in news videos necessitates an interdisciplinary approach, bridging theoretical concepts and findings from communication and media science with automation techniques from computer science. Accordingly, this chapter begins by examining how film editing patterns and narrative strategies shape viewers' perceptions and how they are utilized in news videos. Next, we examine related works from the field of computational video analysis, especially those in the news domain. Lastly, we review works on detecting temporal patterns in news videos.

2.1 News Media Background

News media play an important role in shaping public discourse. Through agenda-setting, news organizations influence which topics dominate public discussions and thus influence societal priorities [53, 74]. Beyond merely relaying information, news also shapes societal beliefs about the covered topics and is fundamental to informed public decision-making and democratic accountability [53, 76, 114]. Despite this critical societal role, the news industry has faced significant challenges in recent years. Some scholars go as far as labeling these challenges a post-truth crisis, characterized by the erosion of traditional journalistic standards and the rise of a politically fragmented, agenda-driven news landscape [114]. In this fragmented landscape, misinformation and disinformation pose substantial threats [8, 28, 32]. Misinformation refers to the unintentional spread of false or inaccurate information, such as reporting errors, while disinformation is deliberately fabricated false information designed to mislead or advance specific agendas [3, 32, 42]. Mis- and disinformation are especially problematic due to the continued influence effect, which describes the perseverance of false beliefs even after misinformation is corrected [66]. A high-profile example is the interference in the 2016 U.S. election, where disinformation campaigns sought to promote narratives favoring Donald Trump [8].

Today, news can be accessed in a variety of formats, ranging from traditional print journalism to short-form videos. In recent years, short and long-form news videos have been a primary source of information for many consumers [81, 82]. According to Newman et al. [81], two-thirds of surveyed individuals consume short-form news videos weekly, and over half of the sample viewed longer formats weekly. Video formats bring unique opportunities and challenges. Research by Lee et al. [65] suggests that consuming news in video form compared to text or image-based news can increase subjects' receptiveness to misinformation [65]. Besides their effects on viewers, news videos also differ from text-based news in how they can be edited. For instance, film editing patterns such as continuity editing facilitate video comprehension by aligning with viewers' natural mental segmentation processes [72]. Furthermore, aligning cuts with viewers natural segmentations' of activities helped increase recall of the presented information [100]. Techniques such as the Kuleshov Effect demonstrate that contextual framing can manipulate audience perceptions of facial emotions [78]. Intensified editing, marked by rapid cuts and dynamic compositions, has become more prevalent, aiming to heighten viewer engagement and immersion [16]. Additionally, narrative strategies in videos help with comprehension [98] and can also increase engagement [29]. Crucially, narrative strategies can also influence the persuasiveness of messages [17, 29].

While narrative strategies and film editing patterns generally spark less debate in cinematic or instructional contexts, their impact on comprehension and persuasiveness takes on heightened significance in the context of news media. The work on analyzing the utilization of narrative strategies and film editing patterns in news videos is limited. Schaefer and Martinez III [99] showed a shift in editing trends over the years with shot length, soundbite length, and realism editing decreasing while montage-type edits and special effects increased in the US between 1969 and 2005 [99]. The decrease in average shot length has also been analyzed for Hollywood films in Bordwell [16] and shows a similar phenomenon. More recently, Bateman and Tseng [13] investigated how state-owned and private broadcasters deploy narrative strategies to meet distinct institutional objectives. Their analysis revealed that while both channels employ such strategies, the public broadcaster typically emphasizes clarity, whereas the private broadcaster favors emotional engagement [13].

Although the presented work lays a valuable foundation, the research remains limited and is often based on small datasets. A likely explanation for this scarcity is the reliance on labor-intensive expert annotations for news videos, which constrains the scalability. To address this challenge, we now turn to computational video analysis, which offers the possibility to conduct news video analysis at scale [120].

2.2 Computational Video Analysis

The field of video analysis has evolved significantly over the past decades, with advancements driven by the increasing availability of video data and compute as well as improved model architectures. Early approaches focused on handcrafted features, while modern methods mostly leverage deep learning architectures to extract features automatically. These methods have also been extensively applied to videos within the news domain.

2.2.1 General Video Analysis

Video analysis has long been a part of computer vision, dating back to early automated approaches in the 1970s and 1980s [1, 58, 71, 80]. Early methods mostly relied on handcrafted features to capture spatial and temporal relationships in videos. To this end, techniques like Scale-Invariant Feature Transform (SIFT) [70, 85], Histogram of Oriented Gradients (HOG) [31], and Optical Flow [14, 58] were used. The derived features were then usually processed using classical machine learning algorithms like Support Vector Machines (SVMs) [27], Hierarchical Decision Trees [90], and Hidden Markov Models (HMMs) [63].

Afterward, in the early 2010s, the acceleration of convolutional neural networks (CNNs) through graphics processing units (GPUs) made CNNs an increasingly popular choice in the field of computer vision. A key milestone in CNN-based image classification was reached by Krizhevsky et al. [64] when they introduced AlexNet. Subsequently, Karpathy et al. [61] showed that the capabilities of CNNs were not limited to the domain of images and could successfully be applied to video classification on a large scale. However, these initial CNN-based video analysis methods typically treated videos as a sequence of independent frames without incorporating temporal relationships. To address this limitation Simonyan and Zisserman [104], introduced two-stream networks. These networks consist of two parallel CNNs that separately process the spatial and temporal information before merging the outputs into a single prediction [104].

Around the same time, researchers combined CNNs with Recurrent Neural Networks (RNNs) and their variants, like Long Short-Term Memory (LSTM) networks. This enabled the processing of longer temporal dependencies [36, 83]. Parallelly, another alternative emerged in the form of three-dimensional Convolutional Neural Networks (3D CNNs), which extended two-dimensional convolutional filters into the time dimension, allowing for simultaneous extraction of spatial and temporal features [110].

More recently, transformer architectures have again initiated a paradigm shift in video analysis. Originally developed for natural language processing [112], transformers have since been successfully adapted to video tasks [15, 69]. Unlike CNNs and RNNs, they use a self-attention mechanism to capture longer temporal relationships[112]. Transformer-based video models such as the Video Vision Transformer (ViViT) [6] and Video Swin Transformer [69] have achieved state-of-the-art performance across multiple tasks like video classification, object tracking, and anomaly detection [6, 69]. Since then, approaches like Video Masked Autoencoders (VideoMAE) have advanced unsupervised pretraining for video representation learning. The key benefit of such approaches is that they do not depend on large annotated datasets [109].

These innovations in video analysis have since been applied to a variety of fields. For example, lane detection, pedestrian recognition, and traffic monitoring in autonomous driving rely on video analysis [44, 49]. In medicine, video analysis offers transformative possibilities by improving diagnostic accuracy [40]. News video analysis also plays a key role in the surveillance domain, for example, through human action recognition [91].

2.2.2 News Video Analysis

Besides the aforementioned application fields, many methods of video analysis have also been applied to the news domain. We will now examine related work from this domain.

A central task in news video analysis is caption extraction, a specialized application of Optical Character Recognition. Xiaoling and Hua [118] examined this task for Chinese news broadcasts, whereas Elagouni et al. [37] applied a neural network-based method for French news videos, attaining 95% character recognition accuracy and correctly recognizing 78% of words. Similarly, Mühling et al. [79] performed Optical Character Recognition on historical broadcasts from the German Democratic Republic's television archive.

Speaker role identification (e.g., news anchor, reporter) is another key challenge in news video analysis. Early studies, like Liu [68], mostly conducted unimodal approaches. Rouvier et al. [97], laid a multimodal groundwork by integrating audio and visual features within a deep neural network. Building on this, Cheema et al. [21] compared multiple models on a more diverse set of speaker roles, finding that although VideoMAE generally performed best on the task, simpler methods like Random Forests (RF) and XGBoost generalized better across different news sources. This finding highlights the trade-off between test performance and real-world adaptability.

Another area of research involves detecting visual concepts in news video frames. For instance, Mühling et al. [79] developed a deep multi-label CNN to recognize domain-specific visual concepts in a German television archive. Their method outperformed previous approaches, achieving an overall average precision of 62.4% and an 83.3% average precision in the recognition of prominent GDR-era individuals, like Erich Honecker.

Another crucial domain in news video analysis is multimodal sentiment analysis. Balahur et al. [12] emphasized the difficulties of sentiment detection in news due to the neutral language of news. Instead, they argued that sentiment in the news context often depends more on context rather than explicit vocabulary. To accurately capture this context in news videos, researchers have experimented with various multimodal approaches. Pereira et al. [89] combined facial emotion recognition, speech modulation, and closed-caption text to achieve 84% accuracy in detecting emotional tension. Kechaou et al. [62] proposed a hybrid hidden Markov model and support vector machine system to classify news videos as "good" or "bad." Further highlighting the value of non-textual information, Ellis et al. [38] found that 21.54% of multimodal sentiment evaluations deviated from text-only analyses. Additionally, Soleymani et al. [105] and Huddar et al. [60] demonstrated that multimodal sentiment analysis outperforms text-only methods, although integrating diverse modalities can be a challenge.

While these studies offer important approaches for tasks like classification and concept extraction, not all findings are applicable to the domain of temporal pattern detection and exploration. Since this thesis centers on detecting specific temporal patterns—particularly narrative strategies and film editing—we now turn our attention specifically to that domain.

2.2.3 Temporal Pattern Detection in News Videos

Temporal segmentation into coherent units, such as stories and shots, has long been a prominent research area in the news domain. Hauptmann and Witbrock [54] introduced an early multimodal approach for story segmentation. Subsequently, Hsu et al. [59] approached this task by applying a maximum entropy statistical model to fuse diverse features from multiple modalities, and Poulisse et al. [93] reported improved performance with a similar framework. Kolekar and Sengupta [63] followed a different approach by utilizing a hierarchical approach and hidden Markov models to segment news into stories. Vinciarelli and Favre [113] also utilized hidden Markov models to detect news stories, but instead of on a content basis, they did so by analyzing the social roles of the speakers.

Beyond story segmentation, shot boundary detection has also received attention from the research community. Cooper et al. [26] approached this task by pairing supervised learning with specialized kernels and pairwise frame similarity measures. They highlighted the positive impact of feature selection on model performance. More recently, Chakraborty et al. [19] utilized principal component analysis for feature extraction and a distance-based algorithm to identify shot boundaries. Then, the authors refined the results with a CNN to achieve high accuracy in detecting both abrupt and gradual shot transitions.

Another task that can be viewed as temporal pattern detection is video summarization, as it usually relies on the detection of highlights within the temporal structure. Many works on video summarization use news videos as part of their evaluation set. Early work by Gong and Liu [47] applied Singular Value Decomposition to cluster visually similar frames into video summaries, though this approach struggled to capture motion. Gygli et al. [51] addressed this limitation by introducing "superframes," which are segments aligned with logical boundaries through a fusion of low-, mid-, and high-level features to quantify "interestingness." Mahasseni et al. [73] approached the task with an unsupervised adversarial LSTM framework to reduce reliance on labeled data. More recently, Ghauri et al. [45] introduced a supervised Multi-Source Visual Attention model that integrates motion and visual cues with an attention mechanism to enhance summaries.

While temporal segmentation and video summarization have been widely explored in the news domain, other temporal patterns, such as film editing techniques and narrative strategies, remain underexplored. Even outside the news context, there appears to be little computational work specifically targeting their detection. This thesis addresses that gap by investigating temporal pattern detection in news videos, focusing on narrative strategies and film editing patterns. Through advanced machine learning techniques, including random forests, transformers and vision-language models, it aims to establish a framework for identifying and analyzing temporal patterns in the news domain. Before outlining the methodology, we will now look deeper into the foundational concepts relevant to this work.

3 Foundations

As outlined in Section 2.2.3, some previous works address the automated detection of temporal patterns in news videos. However, these existing approaches differ from this thesis in two main aspects: they focus on different types of temporal patterns and often use different machine learning methods. This chapter introduces the core concepts and techniques needed to address these differences.

Section Section 3.1 defines the specific temporal patterns examined in this thesis. Film editing patterns refer to shot sequences with distinct framing and relation characteristics, such as depicting alternating shots. Narrative strategies, in contrast, capture higher-level editorial decisions such as the emotionalisation of a topic. Both pattern types are formally defined to support annotation and automated detection. Section Section 3.2 outlines the machine learning methods used for pattern detection. We begin with Vision-Language Models (VLMs), which combine visual and textual inputs via Transformer-based architectures and serve as a baseline. We then introduce tree-based ensemble classifiers—Random Forests and XGBoost—which, although not sequential by design, can be adapted for temporal pattern analysis through feature engineering. Together, these definitions and methods form the theoretical foundation for the analyses in the following chapters.

3.1 Temporal Pattern Taxonomy

This section provides a formal definition of the specific temporal patterns studied in this thesis. We begin by outlining a formalized language for defining patterns in videos, focusing on film editing techniques. These define recurring visual structures at the shot level. We then introduce the narrative strategies, which reflect higher-level editorial or thematic decisions in news storytelling.

3.1.1 Film Editing Patterns

While film editing patterns can seem intuitive to those familiar with the medium, they require formalized descriptions for both scientific rigor and computational analysis. To this end, Wu et al. [117] proposed a language that describes film editing patterns based on framing properties (properties of a single shot) and shot relations (describing transitions between consecutive shots).

Some relevant examples of framing properties include:

- Shot Size: Ranging from wide establishing shots to extreme closeups.
- Number of Actors: The number of actors present on screen.

On the other hand, shot relations describe changes or consistency between consecutive shots, for example:

- Size Relations: Captures whether the shot size stays the same, gets closer, or moves farther away.
- Actor relations: Whether the same or different actors appear in successive shots.
- Place Relations: Whether there is a location change between shots.

To create clear pattern definitions, framing properties and shot relations are combined in a syntax specifying sequences or subsequences of shots with constraints on length, repetition, or variation [117]. Building on this framework, Tseng et al. [111] defined a set of film editing patterns specifically relevant for the news context, such as the pattern `alternating-shots`:

```
alternating-shots {
    length: >= 4
    size-relation: repeating same size
    actor/object-relation: repeating at least 2 sets of different actors/objects
}
```

This definition specifies that an alternating-shots pattern must consist of at least four shots to allow a back-and-forth repetition, must maintain the same shot size whenever it returns to a previous shot, and must involve at least two distinct sets of actors or objects that also repeat. The definition does not specify any framing properties, it depends instead solely on shot relations. Tseng et al. [111] is part of the ongoing research project FakeNarratives [1]. Since the project is still ongoing, the definitions are subject to change and have since been refined and expanded. For instance, the pattern alternating shots was split into four variations centering on specific combinations of actors and objects. Since the number of film editing patterns is vast, their full definitions (as of January 2025) are provided in the supplementary materials. Here, we only display a high-level description of each pattern to enable subsequent discussion.

- Actor Continuity: Focuses on the same actor(s) over several shots.
- Alternating Shots A (Actors): Alternates between two distinct (groups of) actors across multiple shots, with each appearing at least twice.
- Alternating Shots B (Objects): Alternates between two distinct (sets of) objects across multiple shots, with each appearing at least twice.
- Alternating Shots C (Objects + Actor): Cuts from one or multiple object(s) to one or multiple actor(s), then returns to the same object(s) before returning again to the same actor(s).
- Alternating Shots D (Actors + Objects): Cuts from one or multiple actor(s) to one or multiple object(s), then goes back to the same actor(s) and object(s).
- Ambiance Enhancement: Maintains consistent sound or music for at least three consecutive shots.

[1] https://fakenarratives.github.io/

- Continuity of Talk/Dialogue: Preserves dialogue continuity across multiple shots, even if the speaker is not shown.
- Cut-away Version 1 (To People): Momentarily shifts focus from the main subject to a group of people (often for context or reaction) and then returns to the original subject.
- Cut-away Version 2 (To Objects): Temporarily shifts to one or multiple object(s) by inserting a different shot and then returns to the original subject.
- Cut-in: Briefly transitions from a wider shot to a closer view or detail of the same scene (e.g., part of an object), then moves on or returns.
- Double Cut-in: A variant of the cut-in involving two consecutive close-ups of the same object for emphasis.
- Frameshare: Consecutive shots of different actors occupying the same on-screen region, typically to suggest agreement or alignment between them.
- Intensify Version 1 (Same Face): Progressively zooms in on the same face over a sequence of shots.
- Intensify Version 2 (Multiple Faces): Progressively zooms in on multiple faces over a sequence of shots.
- Intensify Version 3 (Object): Progressively zooms in on an object over a sequence of shots.
- Opposition: Consecutive shots of different actors framed in different on-screen regions, often conveying disagreement or contrasting viewpoints.
- Shot Reverse Shot: Repeatedly alternates between two characters (e.g., in conversation), commonly used to depict dialogue or direct interaction.
- Spatial Continuity: Depicts the same location over several shots.
- Thematic Enhancement: Emphasizes a theme by repeating a particular action or movement (e.g., walking, queueing) across multiple shots.

This wide range of film editing patterns shows the diverse editing choices news broadcasts can take to shape their stories and develop a distinctive editorial style. Even when reporting the same factual information, two broadcasts can vary significantly in their impact on viewers due to such editorial choices. In the next section, we examine how these decisions can also serve deliberate narrative goals.

3.1.2 Narrative Strategies

While film editing patterns reflect structural techniques at the shot level, narrative strategies operate at a more thematic or editorial level. These strategies influence how events and actors are framed in the story. Tseng et al. [111] identified six key narrative strategies utilized in news videos: fragmentation, dramatization, emotionalisation, individualization of reporter, individualization of layperson, and individualization of elite. As of January 2025, these strategies have been expanded and formally defined, enabling precise annotation and algorithmic detection.

Individualization of Elite: This strategy focuses on highlighting a single elite individual - someone with power or high public standing - rather than referring to a group or institution (e.g., “the government”). By showing footage of this elite person in a studio context where they are introduced by name, and then following up with a shot of them in a different (non-studio) environment, the usage of this pattern underscores their personal agency or responsibility. Formally, it is defined as:

```
individualisation-of-elite {
    length: >= 2
    shot1: place = studio, talkspace = on-screen,
            spoken_text >= 1 named entity (person)
    shot2: place != studio, talkspace = off-screen,
            actor.role != {reporter, anchor}
    actor-relation: same (all)}
```

Individualization of Layperson:

This strategy parallels the individualization of an elite figure but instead focuses on someone from the general public - a layperson. The viewer typically sees and hears this individual, giving them a face and a voice within the narrative. The layperson can, for example, be someone affected by a specific policy, event, or phenomenon, thereby creating a personal and relatable element in the news story. The strategy individualization of layperson is present when a series of shots depict a layperson in a non-studio context, perhaps in a real-world setting such as a home or workplace, which underscores the everyday nature of the subject’s experience and sets them apart from professional media figures.

```
individualisation-of-layperson {
    length: >= 2
    place: != studio
    last_shot: actor.role != {reporter, anchor}, talkspace = on-screen
    actor-relation: at least one actor same (all)}
```

Individualization of Reporter: When this pattern occurs, the focus lies on a member of the media in a situation outside of the news studio. By making a self-reference (e.g. "**My** opinion is..."), the reporter is no longer just an informational medium and becomes a personalized, possibly emotionally involved figure.

```
individualisation-of-reporter {
    length: >= 1
    last_speaker_turn: spoken_text >= 1 self-referent-pronoun,
    last_shot: talkspace = on-screen
    place != studio,
    actor.role = reporter
```

Emotionalisation: Occurrences of this pattern heighten the emotional tension of a topic. By displaying the same emotion (e.g., anger, sadness, excitement) across multiple shots, the usage of this pattern highlights that emotion's importance to the narrative. These strategies are meant to evoke a similar emotional reaction in the viewer.

1. Emotionalisation (Facial): Is realized when a non-neutral face emotion is depicted for multiple subsequent shots.

    ```
    emotionalisation-facial {
        length: >= 2
        face-emotion: not neutral
        face-emotion-relation: same (all)}
    ```

2. Emotionalisation (Sentiment): This version focuses on the general sentiment of speaker turns. It can be realized in two ways. Either the sentiment in one speaker turn is high in some emotion, or the sentiment is simply not neutral over multiple speaker turns.

    ```
    emotionalisation-sentiment {
        length: = 1 speaker-turn
        sentiment: highly positive/negative
    ```

    ```
    emotionalisation-sentiment {
        length: >= 2 speaker-turns
        sentiment: not neutral (all)
        sentiment-relation: same (all)}
    ```

Dramatisation: The pattern Dramatisation occurs when a reporter is actively moving (e.g., walking or running) while speaking on-screen, and being followed by the camera. This makes the report more dynamic and can create a sense of urgency.

```
dramatisation {
    length: = 1
    actor.role: reporter
    talkspace: on-screen
    actor.movement: walking or running
    camera.movement: with actor}
```

Fragmentation: Usage of this pattern splits visual and/or verbal information to highlight specific story elements or multiple perspectives. It appears in three versions:

1. Fragmentation with Inserted Shots: Shots not displaying the speaker are alternated between shots of the same speaker. This creates a break in visual continuity:

    ```
    fragmentation-with-inserted-shots{
        length: >= 4
        speaker-relation: same (all)
        shot1 + shot3: actor-relation = same, place-relation = same,
                talkspace = on-screen
        shot2 + shot4: object-relation = same, place-relation = same,
                talkspace = off-screen
        shot1 + shot2: visual-similarity = low}
    ```

2. Split-Screen Fragmentation: Two or more visuals appear simultaneously in a divided frame. This fragments the viewer's attention within a single frame.

```
fragmentation-with-split-screen {
    length: = 1
    actor.role: != reporter,
    talkspace: on-screen,
    splitscreen: true}
```

For fragmentation, we can observe the connection between film editing patterns and narrative strategies. Fragmentation with inserted shots essentially relies on a more constrained occurrence of the FEP alternating shots D. As discussed in Chapter 2, the utilization of narrative strategies and film editing patterns has an impact on how viewers perceive the displayed information. Editors can highlight certain story elements or guide audience reactions in deliberate ways. Recognizing and examining these techniques at scale is thus essential for understanding the possible agendas of news reporting. In the next section, we turn to machine learning methods that enable such a detection of temporal patterns at scale.

3.2 Machine Learning Methods for Pattern Detection

With a clear understanding of the target patterns, we now introduce the computational methods used to detect them automatically. We begin with vision-language models (VLMs), which serve as a zero-shot baseline. Next, we present two tree-based classifiers—Random Forests and Gradient Boosting Classifiers—which form the main predictive models used in this study.

3.2.1 Transformer-Based Vision-Language Models

Recent advances in multimodal learning and generative AI have been largely driven by the Transformer architecture [112], which has revolutionized sequence modeling by focusing on (self)-attention mechanisms [9]. The core idea of self-attention is to let each token in the sequence assign different levels of importance (or "attention") to all the other tokens, thereby capturing contextual dependencies regardless of their distance. For this, each word has three weights assigned to it, one being the query, another the key and a third being the value. Query (Q): Represents the "question" a token is asking—what information does this token need? Key (K): Represents the "address" or "label" of each token—what is this token offering? Value (V): Represents the content or information carried by each token—the actual information to be passed along once selected via the Key-Query match [112].

Recent advancements in multimodal learning and generative AI have been largely driven by the emergence of the Transformer architecture [112], which revolutionized sequence modeling through the utilization of (self-)attention mechanisms [9]. The central idea of self-attention is to allow each token in a sequence to assign varying degrees of importance to every other token, thereby capturing contextual dependencies regardless of their position in the sequence. To achieve this, each token is associated with three learned representations: a Query (Q), a Key

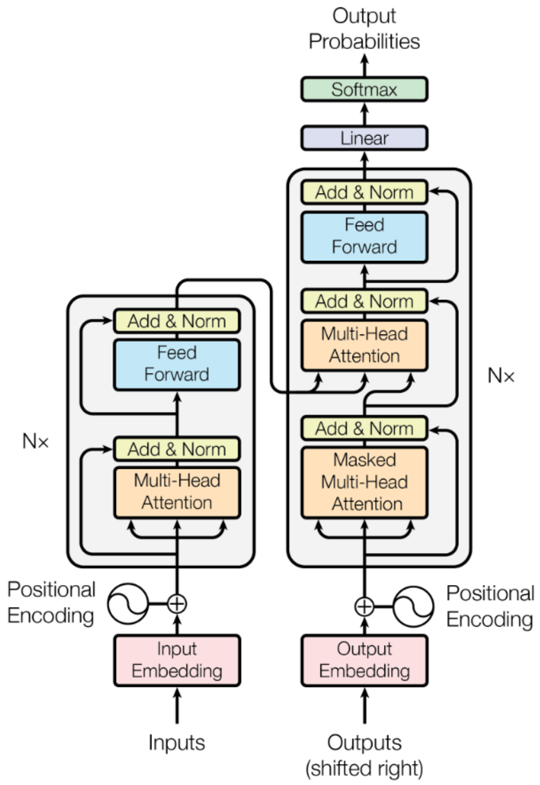


Figure 3.1: Transformer architecture. Reproduced from Vaswani et al. [112].

(K), and a Value (V). When processing a sequence, each token takes the role as the query at some point. The query represents what information a token is seeking. This query is multiplied with the keys of the other tokens, these keys represent the information a token contains. The dot product $QK^\top$ thus measures the similarity between each query and key. After determining through this mechanism which tokens are relevant for each other, the values V that hold the actual information are multiplied. Formally, in Vaswani et al. [112] attention is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

where Q, K, and V are matrices for the Query, Key and Value while d_k is the dimensionality of the key vectors. $d_k$this is used to scale the result of $QK^\top$ while the softmax function normalizes the result.

As depicted in Figure 3.1, Transformer models typically consist of blocks, each containing:

1. Multi-Head Self-Attention: Instead of computing a single set of Q, K, and V vectors, the Transformer computes multiple sets of attention (called heads) in parallel. This allows the model to capture multiple different types of relations between the tokens. The outputs of all heads are then concatenated and combined with previous information through a residual connection.
2. Feed-Forward Network: Each token's representation is further processed by a feed-forward network applied independently at each position.

This stands in contrast to earlier architectures such as recurrent neural networks (RNNs) [39] and long short-term memory networks (LSTMs) [57], which process inputs sequentially, are thus not parallelizable and often struggle to retain information over very long sequences due to vanishing gradients [56]. In contrast, transformers allow for more parallelization and can model long-term relationships more effectively [112]. These properties have enabled breakthroughs in natural language processing through the rise of large language models (LLMs), vision through Vision Transformers (ViTs), and more recently, the integration of both modalities in *Vision-Language Models* (VLMs), which aim to jointly process and reason over visual and textual inputs. Notable developments over the past years in this area include:

- VideoBERT [108] applies masked language modeling to learn joint representations of video and text.
- CLIP [95] uses separate encoders for images and text, aligning their output embeddings in a shared latent space via contrastive learning.
- ViViT [6] extends the Transformer architecture for spatiotemporal video analysis.
- Flamingo [4] combines pretrained vision-only and language-only models, enabling training on any data that includes images and/or text.
- BLIP-2 [67] reduces training costs by using frozen, pretrained vision and language models, bridging the modality gap with a querying transformer.

Regarding the current state of the art, both proprietary and open-access models are evolving rapidly. Current prominent proprietary models include GPT-4o [86], Gemini 2.0 Flash [77], and Claude Sonnet 3.5 [5], while open models include LLaMA 3-V 405B [75], InternVL2 [87], NVLM [30], Molmo [34], Pixtral-12B [2], and most importantly for this work Qwen2.5-VL [10].

As this work utilizes **Qwen2.5-VL** [10], we briefly describe its architecture. The model consists of three core components:

1. A Vision Transformer (ViT) encoder that embeds input images and videos of varying sizes and frame rates
2. A large language model initialized with weights from the LLM Qwen2.5 [119].
3. A multi-level perception-based merger that compresses visual features into the language embedding space.

Additionally, in all but four layers, Qwen2.5-VL replaces standard self-attention with window-based attention, which computes attention locally rather than across the entire visual sequence. Since the computational complexity of attention grows quadratically ith the input size and vision inputs tend to be high dimensional, this modification reduces computational complexity, particularly for high-resolution inputs. To retain the spatial structure of visual data, Qwen2.5-VL employs **Multimodal Rotary Positional Embeddings**—an extension of Rotary Positional Embeddings [107]—which encode the relative position of visual patches in two dimensions. This allows the model to reason about layout and spatial relationships in documents, scenes, and diagrams. An additional focus in creating Qwen2.5-VL was thenprocess of data selection and cleaning during training, as data quality significantly impacts model performance. After the initial training, the model is fine-tuned using human feedback to enhance alignment and task performance [10].

3.2.2 Ensemble Classifiers

Unlike VLMs, which learn high-level representations from raw visual and textual inputs, ensemble classifiers such as Random Forests and Gradient Boosting Classifiers rely on hand-engineered features. In the following we introduce two different ensemble classifiers used in this work.

3.2.2.1 Random Forest Classifier

Random forests are a popular machine-learning method that builds on decision trees. Through ensemble learning, they combine the decisions of multiple decision trees to reduce errors. Random forests can be applied to both classification and regression tasks. In their original formulation, a class is predicted through a simple majority vote among the trees while a regression value is calculated as the average of the predicted values [18]. However popular current implementations now use the average probabilistic predictions of the single trees in classification tasks as well [88]. The training of a random forest begins with bootstrapping, where random samples are drawn from the training data (with replacement) to create multiple subsets. On each of these subsets a decision tree is subsequently trained. Bootstrapping is not the only way randomization is introduced into the training process, feature randomization can also be employed to further diversify the trees. This means that at each decision within the trees, only a randomly selected subset of features is considered for splitting, rather than evaluating all available features [18, 55]. Through the deliberate introduction of randomness, random forests are less prone to overfitting and are known for their good generalization capabilities. Additionally, random forests are scalable and can efficiently handle large datasets with numerous features. They are also inherently non-parametric, requiring no assumptions about the underlying data distribution, which contributes to their versatility. Another significant advantage is their ability to provide feature importance scores, offering insights into which variables most influence the predictions [18].

However, random forests are not without limitations. The training process of random forests with many deep trees can be computationally expensive, especially for large datasets. Moreover, while feature importance scores enhance interpretability, the overall model is regarded as a "black box" compared to simpler models like individual decision trees [**molnar**]. The randomized selection of features can also lead to underfitting in some cases. Most importantly for this work, random forests are not inherently suitable for detection tasks as they can not directly process temporal sequences. They can, however be adapted for such tasks, for example, by aggregating data through a sliding window approach and treating the created instances as a classification problem. Random forest have also been applied in two related tasks. Hsu et al. [59] combine multimodal features and a decision tree as a part of their system to predict story boundaries. Cheema et al. [21] have used random forests in a combination with multimodal features to classify speaker turns and news type situations and highlighted their generalization capabilities in these tasks.

3.2.2.2 Gradient Boosting Classifier

Gradient boosting is a tree-based ensemble method that builds models sequentially, with each new tree trained to correct the residual errors of the current ensemble's predictions [43]. Rather than relying on randomness, as in bagging-based methods, gradient boosting takes a more directed approach: each step minimizes a specified loss function by fitting the negative gradient of that loss. This allows the model to focus learning on the most challenging examples, effectively performing greedy function approximation.

A notable implementation of gradient boosting is XGBoost [24], which introduces regularization terms to its objective function to prevent overfitting and improve generalization. XGBoost is also engineered for performance, supporting parallel processing, missing value handling, and optimized memory usage, making it well-suited for large-scale and heterogeneous datasets. However, like other tree-based models, gradient boosting does not inherently capture temporal dynamics; any temporal structure must be encoded explicitly through feature engineering. Despite this limitation, gradient boosting methods—particularly when paired with well-crafted input features—have shown strong predictive capabilities across a wide range of tasks, including news video analysis [21].

4 Methodology

This chapter presents the methodology used to address the research gap identified in Chapter 1. We begin by formally defining the detection and exploration tasks in Section 4.1, introducing spans as the basic processing units that encompass both visual shots and speaker turns. Section Section 4.2 describes the data preprocessing pipeline, including the conversion of annotations and the extraction of human-interpretable features. These processing steps transform raw news video data into structured, span-level instances suitiible for our models.

Section Section 4.3 outlines two approaches for temporal pattern detection ranging from model inputs to model outputformats. The first is a supervised method using tree-based classifiers that aggregate features across sliding windows of spans to predict the presence of editing patterns or narrative strategies. We implement this approach both in a multi-label setting and a per-class setting, addressing data imbalances in the latter with SMOTE. The second approach uses a zero-shot Vision-Language Model (VLM) baseline, which processes visual frames and transcripts within a five-shot context window, also in both multi-label and per-class setups. Finally, Section Section 4.4 briefly describes the validity checks applied throughout the pipeline to ensure robustness and reproducibility.

4.1 Problem Formalization and Task Overview

This section formalizes the problem setting. We begin by introducing the concept of spans, which serve as the fundamental processing units of our methodology. We then present the main objectives Film Editing Pattern Detection (FEPD), Narrative Strategy Detection (NSD) and Pattern Exploration.

4.1.1 Definition and Motivation for Spans

Our approach processes videos at the span level. A span encapsulates the visual information of a single shot as well as any speaker turns occurring during that shot. If a speaker turn extends over multiple shots, its complete speaker-turn data is included in each relevant span. Consequently, consecutive spans may have identical textual/audio content but differ in their visual information. However, every individual span always corresponds to exactly one shot in terms of both visual data and target information. The span was chosen as the processing unit since aggregating all features at the shot level can be problematic if a shot is very short. For instance, if we want to analyze a speaker's sentiment, but the shot only contains a single word, such features may be uninformative. A schematic illustration of spans is shown in Figure 4.1. Formally, given a news video V with shot boundaries, we define a sequence of contiguous spans:

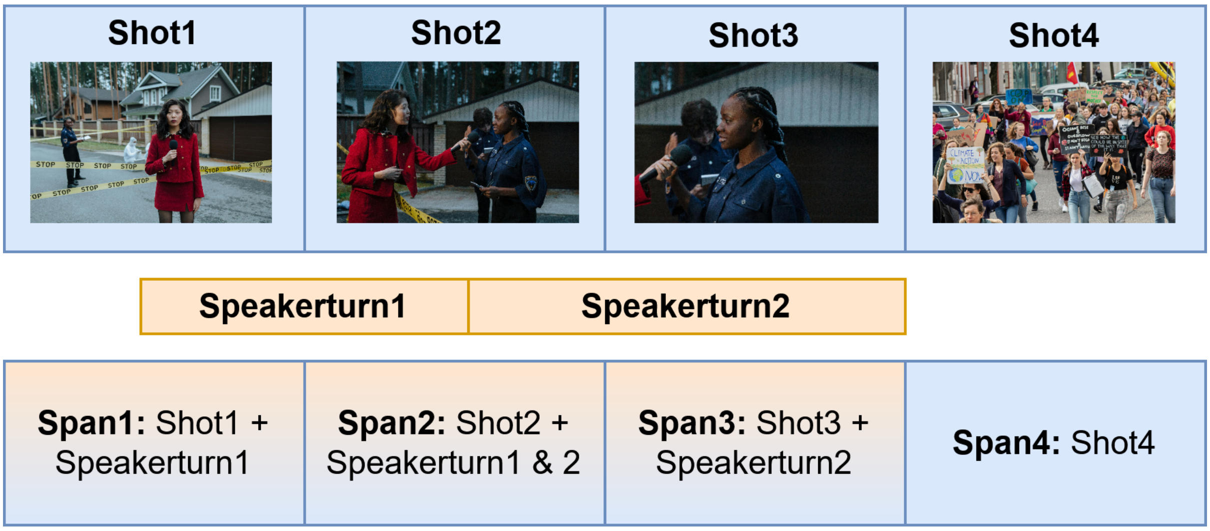

Shot1
Shot2
Shot3
Shot4
Speakerturn1
Speakerturn2
Span1: Shot1 + Speakerturn1
Span2: Shot2 + Speakerturn1 & 2
Span3: Shot3 + Speakerturn2
Span4: Shot4

Figure 4.1: Illustration of the concept of spans and how they relate to shots and speaker turns.

$$\mathbb{S} \;=\; \{\, S_1,\, S_2,\, \ldots,\, S_n \},$$

where n denotes the total number of spans, and the union of all spans forms the video:

$$\bigcup_{i=1}^{n} S_i \;=\; V.$$

4.1.2 Detection and Exploration Tasks

Our primary goal is to detect and analyze specific temporal patterns in news videos. We focus on two main detection tasks – Film Editing Pattern Detection and Narrative Strategy Detection – and one exploration task.

Film Editing Pattern Detection Most videos contain edits, meaning they consist of more than one continuous shot. The order in which editors present shots is normally not due to random chance but rather a deliberate choice. Drawing from previous work [111] and expertise from communication scientists in the ongoing project Fake Narratives [1], we define a set of film editing patterns:

$$\mathbb{C}_{FEP} = \{\, F_1, F_2, \ldots, F_n \},$$

as outlined in Section 3.1.2 and detailed in Appendix A.1.

Given a video V, the goal is to detect all film editing patterns $F \in \mathbb{C}_{FEP}$ that occur during the video. Formally, we define a ground-truth labeling function

$$\Theta_{\text{FEPD}} : S_i \;\longrightarrow\; \mathcal{P}\big(\mathbb{C}_{FEP}\big),$$

[1] https://fakenarratives.github.io/

Which means, for each video span S_i, the function Θ_{FEPD} returns a subset of all possible film editing patterns $\mathbb{C}_{FEP}$. This subset contains all patterns present in S_i and could also be the empty set if the span is not a part of any pattern. We define the set of spans in which F_j is present as:

$$\Omega_{F_j} \;=\; \bigl\{\, S_i \in \mathbb{S} \,\big|\, F_j \in \Theta_{\text{FEPD}}(S_i) \bigr\}$$

Next, let

$$\Psi_{\text{FEPD}} : S_i \;\longrightarrow\; \{0,1\}^{|\mathbb{C}_{\text{FEP}}|}$$

be a machine learning model that, given a span S_i as input, outputs a binary prediction for each pattern in $\mathbb{C}_{\text{FEP}}$. Concretely, $\Psi_{\text{FEPD}}(S_i, F_j) = 1$ indicates that span S_i is predicted to contain the film editing pattern F_j, and $\Psi_{\text{FEPD}}(S_i, F_j) = 0$ otherwise. The set of spans in which F_j is predicted to be present is then

$$\widehat{\Omega}_F \;=\; \bigl\{\, S_i \in \mathbf{S} \,\big|\, \Psi_{\text{FEPD}}(S_i, F_j) = 1 \bigr\}.$$

Given these definitions, a perfect solution for the film editing pattern detection task requires that the predicted occurrences align exactly with the true occurrences for each pattern:

$$\forall\, F_j \in \mathbb{C}_{\text{FEP}} : \quad \widehat{\Omega}_{F_j} \;=\; \Omega_{F_j}.$$

Narrative Strategy Detection When multiple editing choices are collectively aimed at creating a specific affect in the viewers, for example, heightening their emotional state, a narrative strategy is being employed. The specific set of narrative strategies $\mathbb{C}_{NS}$ considered in this work are defined in Section 3.1.2. Analogous to FEPD, given a video V, the goal of the second task is to detect all narrative strategies $N \in \mathbb{C}_{NS}$ that occur during the video. Formally, we define the ground-truth labeling function

$$\Theta_{NS} : S_i \;\longrightarrow\; \mathcal{P}\bigl(\mathbb{C}_{NS}\bigr),$$

which, for each video span S_i, returns a subset of all possible narrative strategies $\mathbb{C}_{NS}$. This subset contains every strategy present in S_i and can be empty if no strategy is employed in that span. We denote the set of spans in which N is truly present by

$$\Omega_{NS_j} \;=\; \bigl\{\, S_i \in \mathbb{S} \,\big|\, NS_j \in \Theta_{NS}(S_i) \bigr\}.$$

Additionally, let

$$\Psi_{\text{NS}} : S_i \;\longrightarrow\; \{0,1\}^{|\mathbb{C}_{\text{NS}}|}$$

be a machine learning model that takes a span S_i as input and outputs a binary prediction for each narrative strategy in $\mathbb{C}_{\text{NS}}$. Then $\Psi_{\text{NS}}(S_i, NS_j) = 1$ indicates that span S_i is predicted to contain the film editing pattern NS_j, and $\Psi_{\text{NS}}(S_i, NS_j) = 0$ means that the pattern NS_j was not detected in S_i. The set of spans in which NS_j is predicted to be present is then

$$\widehat{\Omega}_{NS_j} \;=\; \bigl\{\, S_i \in \mathbf{S} \,\big|\, \Psi_{NS}(S_i, NS_j) = 1 \bigr\}.$$

A perfect solution for the Narrative Strategy Detection task thus requires:

$$\forall\, NS_j \,\in\, \mathbb{C}_{\mathrm{NS}} : \quad \widehat{\Omega}_{NS_j} \;=\; \Omega_{NS_j}.$$

Pattern Exploration Beyond the formal detection tasks, this work also involves exploring film editing patterns and narrative strategies by examining their overall frequency, their occurrence by news source, their co-occurrence with one another, and their correlations with various features. These experiments, discussed in **??**, were conducted using Python and pandas.

4.2 Data Preprocessing Pipeline

Having defined the key tasks and formalized the notion of spans, we now turn to the data preprocessing stage. Data is at the heart of machine learning. In this thesis, data preprocessing plays an integral role due to the high-dimensional nature of video data. Working with raw video data can make model outputs difficult to interpret. Furthermore, training models with high-dimensional data necessitates a substantial amount of data due to the curse of dimensionality. To avoid these pitfalls, we extract interpretable features from the raw video data and train our models on these features. The overall preprocessing pipeline—from raw news videos to structured span-level representations—is illustrated in Section 4.2. This section follows the structure of the pipeline and describes each step in detail, including annotation processing, feature extraction, span and story aggregation. Step four of the pipeline, they concrete model inputs, will be discussed when the specific detection models are detailed in Section 4.3.

4.2.1 Annotations

Given a dataset $\mathbb{D}$ of news videos and annotations in the ELAN format [115], with exact start and end times for news stories and temporal patterns, we convert the annotations via a python script into a JSON Format. At this stage, we also map infrequent pattern-subcategories (such as subcategories of alternating shots) into a single pattern to reduce class imbalances. Furthermore the ELAN files also contain shot boundaries detected with the TransNet-V2 model [106], which serve as the basis for later processing. Based on these shot boundaries, for each annotated news story, we aggregate the temporal patterns on the span level. Since the discussed patterns by definition always begin and end at shot boundaries, this simplifies processing without compromising validity.

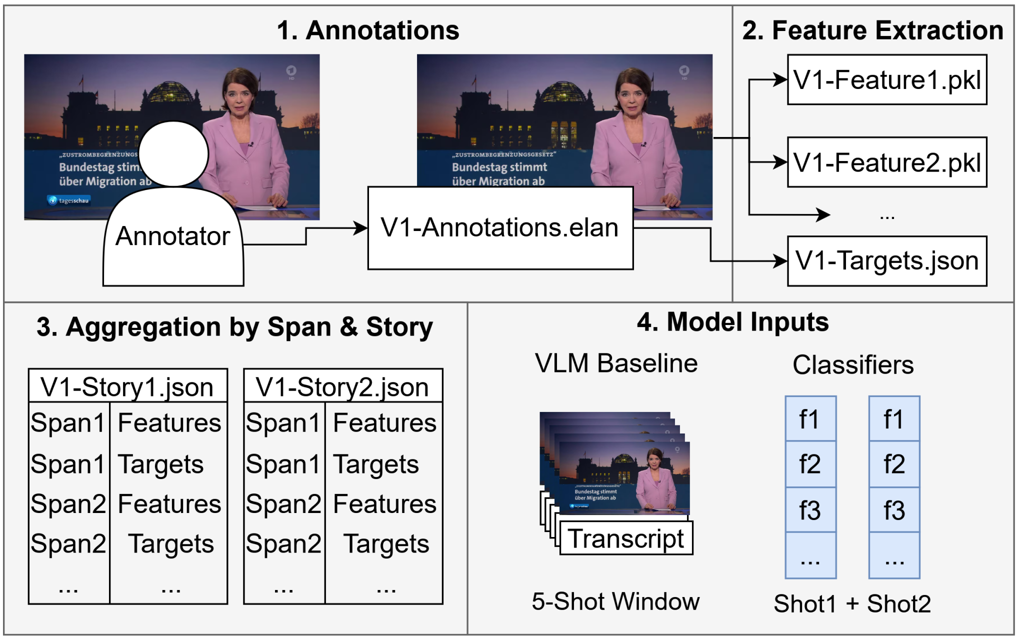


Figure 4.2: Overview of the data processing pipeline. The pipeline consists of four main steps: (1) expert annotation of news stories (2) extraction of multimodal features and pattern labels from raw video and annotation files; (3) aggregation of features and labels by span and story; (4) construction of input data for vision-language models and classifiers.

4.2.2 Feature Extraction Overview

This section describes how we extracted human-interpretable features from raw video data. The base features were obtained as part of the FakeNarratives project[2]. We describe the features used, briefly outline their extraction, and focus on how they were further processed for this thesis.

The selection of features was informed by a multitude of reasons, some features were already present as part of FakeNarratives, others depended more on concrete definitions. in general the feature vector for classifers contains 81 dimensions and there are two additional features for vlms, for many of them thresholds had to be set. as we will later discuss hyperparametr tuning, there was a lot that had to be decided without prior work to draw from and values were picked based on observing the data but not specifically validated or tested. Because this work leverages both classifier and vision-language models, we discuss feature inputs relevant to both approaches.

Feature selection was informed by a combination of practical, conceptual, and technical considerations. Some features were directly inherited from the FakeNarratives project, while others were further engineered based on their theoretical alignment with the pattern definitions central to this study. The final feature vector used in classifier-based models consists of 81 dimensions, with two additional features extracted for use with vision-language models (VLMs). Throughout the pipeline, numerous decisions were required, ranging from thresholding values to aggregation

[2] For more information, see: `https://github.com/FakeNarratives/fakenarratives/tree/main`(status: March 2025)

strategies. In many cases, there was no clear precedent or gold standard to follow, and choices had to be made based on exploratory analysis or practical constraints. While every effort was made to ensure transparency and consistency, it is not feasible to detail the full rationale behind every individual parameter or threshold. Instead, this section aims to clarify the general logic behind the feature design and highlight the most relevant processing steps for reproducibility and interpretation.

4.2.2.1 Vision-Language Model Input Features

As part of our methodology, we utilize a vision-language model. Such models usually take natural language text and/or images as input. In our case, we derived audio transcripts from the videos using the WhisperX model [11]. Besides being used as an input for the VLM, these transcripts also serve as the basis of the text-based features described in Section 4.2.2.2. Furthermore, we iterate over each video to extract the middle frame of each shot, using it as the representative image for that shot. We then convert these images into the base64 format and save them with the accompanying transcript for that span in the JSON format.

4.2.2.2 Classifier Input Features

While VLM features are relatively straightforward and limited in number, traditional classifiers require more extensive feature preprocessing. Here, we detail the features extracted for the classification models.

Visual features

Shot Scale Movement: Each shot is classified as an extreme close-up (ECS), close-up shot (CS), medium shot (MS), full shot (FS), or long shot (LS). For this classification, a version of VideoMAE [109] fine-tuned on the MovieNet dataset was employed. For the shot-scale task, this model achieves an accuracy of 88.32% and a macro-F1 of 88.57% [22]. In this thesis, these ordinal categories were encoded from 0 (ECS) to 1 (LS) in increments of 0.25. This allowed for a naturally scaled representation of shot scale and made it simple to later on average the feature for classifier input instances (see Section 4.3.1).

Shot Density: Shot boundaries are detected using the TransNet-V2 model [106], which achieves an F1 score between 77.9% and 96.2%, depending on the dataset. Based on these shot boundaries, normalized shot density values between 0 (low density) and 1 (high density) are computed for every 0.04-second interval. As part of this thesis, these values were averaged across each span.

Face Analysis: We detect each face and its bounding box using DeepFace [101, 102]. Based on these detections, we derived the following features on a span basis:

- Unique Faces: The number of distinct faces appearing in a span, identified by unique face IDs.
- Average Face Size: The mean size of all detected faces within a span.

- Region upper/lower: Whether each face is in the upper or lower half of the screen, averaged over the span.
- Region left/right: Whether each face is in the left or right half of the screen, averaged over the span.

Facial Emotions: Emotions are also extracted with DeepFace [103]. The emotion classes are angry, disgust, fear, happy, sad, surprise, and neutral. For this thesis, if the predicted probability of any emotion exceeds 0.5 at some point during a given span, it is one-hot encoded, allowing multiple emotions per span. In the absence of faces or strong emotion signals, the vector remains all zero.

Shot Similarity Features: Various visual similarity metrics were computed, each containing values for the two preceding shots and the two subsequent shots. Two versions of shot similarity were implemented: one using ConvNeXt-v2 [116] and another using SigLIP [121]. Additionally, action similarity was computed via three models: a version of X-CLIP [84] trained fully supervised on Kinetics-600, a VideoMAE model fine-tuned on Kinetics-400 [109], and a second VideoMAE model fine-tuned in a supervised manner on Something-Something-v2 for action similarity. Lastly, place similarity was computed using the Places365-CNN [122]. For this thesis, we always selected the mean similarity to the reference shots.

Audio Features Audio Classification: Audio data are classified using the *extract_features* functionality of the BEATs model [23]. Although the model outputs probabilities for 26 audio classes, for this thesis, we map them to nine categories: Speech, Narration, Music, Animal, Vehicle, Siren, Other Sound, Silence, and Artillery Fire. These categories were defined based on logical groupings and observed occurrences. We apply one-hot encoding: if a category's probability exceeds 0.5 at any point in a span, that category is deemed present.

Speaker Gender Detection: For speaker gender classification, we use a version of Facebook's Wav2Vec2-XLS-R [7] fine-tuned on LibriSpeech [41]. The fine-tuned model achieves an F1 score of 99.93% on the test set. These probabilities are one-hot encoded with a threshold of 0.5 per span, indicating whether any male or female speaker is present.

Sentiment Analysis: Transcripts of speaker turns are categorized as negative, neutral, or positive using a version of Google's BERT architecture [35] trained on extensive German language samples [50]. The model's outputs are mapped to the range of 0 (negative), 0.5 (neutral), and 1 (positive) for this thesis and then averaged per span.

Evaluative Scores: We use Qwen-2.5 Instruct with 32 billion parameters [119] via Unsloth [33] to determine whether the transcript contains evaluative talk. The model is prompted to output a prediction (evaluative or not) with a confidence level of None, Low, Medium, or High. For this thesis, these levels are mapped to a scale of 0 (non-evaluative with high confidence) to 1 (evaluative with high confidence). A confidence of None corresponds to 0.5, while Low and Medium map to 0.4 and 0.6, respectively. For multiple speakers, these scores are averaged.

Named Entity Recognition: We recognize named entities categorized as eper, lper, loc, org, event, or misc using Stanza [94]. We then collect the frequency of each category during the course of the span.

Part-of-Speech (PoS) Tags: Again using Stanza [94], we count occurrences of 14 PoS tags—adj, adp, adv, aux, conj, det, intj, noun, num, part, pron, propn, verb, and other. Conjunction types and punctuation are consolidated into the other category for simplicity.

Multimodal Features Speaker Information: Based on the previously extracted face data and speaker turn data, the multimodal features **Active Speaker** and **Unique Speakers** are computed. Active Speaker indicates whether the speaking person is shown on screen. In this thesis, we count how often the speaker is active during a span and divide by the total number of speakers during the span; thus, this feature is the fraction of the span in which the speaker is visible. We also compute the count of unique speakers using speaker IDs, following the same procedure as for unique faces.

CLIP Based: We analyze speaker turns in a multimodal fashion using CLIP [95], a vision-language model pre-trained on large-scale image-text data. We derive features from predefined textual queries for **Speaker Role Identification** and **News Situations**. For Speaker Role Identification, fifteen roles are queried, then mapped to five overarching roles: anchor, reporter, expert, layperson, and elite. For each speaker turn, we take the top predicted role, increment its count, and normalize by the number of speaker turns in the span. Similarly, for News Situations, CLIP is queried to classify the situation as interview, talking-head, speech, commenting, or voice-over. These predictions are processed in the same way.

4.2.3 Aggregation by Story

As described in the previous section, all features were initially aggregated at the span level. In this step, we further organize the data by grouping spans according to individual news stories. Specifically, each span's features and corresponding target labels are combined into structured JSON files, one per news story. This organization enables more flexible dataset splitting and analysis across different stories and sources rather than treating entire videos as single units. An illustration of this step is provided in Section 4.2.

4.3 Pattern Detection

To address the tasks defined in Section 4.1, we train classifiers using the sliding-window instances. For comparison, we also prompt a Vision-Language Model for pattern detection.

4.3.1 Sliding-Window Aggregation for Classifiers

All features were initially aggregated at the span level, as described in Section 4.1.1. Next, we collected the minimum and maximum values and scaled all classifier features to a 0–1 range, making them more suitable for a variety of models. To apply classification models for the detection tasks, we further aggregated these scaled feature vectors into sliding-window instances.To reduce hyperparameters, we fixed the window size at two with a step size of one. Specifically, let $\{ S_1, S_2, \ldots, S_n\}$ be the sequence of spans. We define

$$\mathcal{I} \;=\; \bigl\{ I_1, I_2, \ldots, I_m \bigr\},$$

where each I_i is a sliding window of size two, with a step size of one:

$$I_i \;=\; \bigl(S_i,\, S_{i+1}\bigr), \qquad 1 \,\leq\, i \,<\, n.$$

In other words, I_i aggregates spans S_i and S_{i+1}, thus creating a new instance whose features are derived from these two consecutive spans. Because the shot relation features provide information about previous and following shots, even with this small window size, each instance in the middle of the video includes information about six spans. Instances at the beginning or end of a video still include information from at least four spans (as long as enough spans exist). For consistency, each feature is aggregated across a window in the same manner it was aggregated within each span. That is, if a feature was originally averaged within a span, we also average it across the two constituent spans in I_i. Conversely, if a feature was maximum-aggregated within a span, we take the maximum value over the two spans in I_i.

4.3.2 Supervised Classifier-Based Detection

The input to our classification models is the set of sliding-window instances constructed in Section 4.3.1. Each classifier outputs a probability vector indicating the likelihood that a given instance contains at least one occurrence of each film editing pattern or narrative strategy. Because not all patterns and strategies in $\mathbb{C}_{\text{FEP}}$ and $\mathbb{C}_{\text{NS}}$ appear in our dataset—and some are merged, as discussed in Section 4.2.1—we represent the classifier outputs as 5-dimensional vectors for film editing pattern detection and 6-dimensional vectors for narrative strategy detection. Formally, we define two classification functions:

$$\Phi_{\text{FEPD}} : I_i \;\longrightarrow\; [0,1]^5 \quad \text{and} \quad \Phi_{\text{NSD}} : I_i \;\longrightarrow\; [0,1]^6,$$

where each I_i is a sliding-window (see Section 4.3.1). The j-th component of $\Phi_{\text{FEPD}}(I_i)$ represents the probability that at least one shot in I_i contains the jth film editing pattern. Similarly, $\Phi_{\text{NSD}}(I_i)$ outputs probabilities for each narrative strategy.

In addition to the multi-label classifiers, we also implemented per-class binary classifiers. Specifically, for each film editing pattern $F_j \in \mathbb{C}_{FEP}$, we train a classification function

$$\phi_{F_j} : I_i \;\longrightarrow\; [0,1],$$

which outputs the probability that instance I_i contains the film editing pattern F_j. By concatenating these per-class outputs, we obtain a combined predictor:

$$\widetilde{\Phi}_{\text{FEPD}}(I_i) \;=\; \bigl(\,\phi_1(I_i),\; \phi_2(I_i),\; \ldots,\; \phi_5(I_i)\bigr).$$

An analogous approach is used for narrative strategies, yielding separate binary classifiers ϕ_{NS_j} for $NS_j \in \mathbb{C}_{NS}$ and a corresponding combined predictor $\widetilde{\Phi}_{\mathrm{NSD}}(I_i)$.

A major advantage of this per-class approach is that each classifier is trained on a binary target, which is well supported by standard libraries and methods for addressing data imbalance. In contrast, multi-label learning is less explored, and many existing libraries do not fully support it. By reducing the problem to multiple binary tasks, we were able to apply SMOTE [20] for oversampling and synthetic data generation techniques, thereby helping to mitigate class imbalance issues.

4.3.3 Input Format for Vision-Language Models

As mentioned, for the VLM we obtain a text transcript and a base64 image for each span. A VLM does not require additional features like PoS tags, as it can infer them on its own. To provide sufficient temporal context, each span S_i is evaluated together with its two preceding spans $\big(S_{i-2}, S_{i-1}\big)$ and its two subsequent spans $\big(S_{i+1}, S_{i+2}\big)$. Concretely, we define a five-shot context window centered on S_i as

$$W_i \;=\; \big(S_{i-2}, S_{i-1}, S_i, S_{i+1}, S_{i+2}\big),$$

where each S_i contains both a visual frame extracted as detailed in Section 4.2.2 and the corresponding transcript text for one span.[3]

4.3.4 Zero-Shot Detection Using Vision-Language Models

Vision-Language Models have demonstrated strong performance on a wide range of multimodal tasks without relying on labeled data. This is especially advantageous in our context, where compiling large annotated datasets for film editing patterns and narrative strategies relies on experts and can thus be expensive. Consequently, we compare our supervised approaches with zero-shot VLM-based methods that require no labeled data. The model input is a window W_i as detailed in Section 4.3.3 combined with a prompt P to guide the model's response. To investigate the impact of prompt formulation, we experiment with two categories of prompts: multi-class and per-pattern binary. Multi-class prompts request the detection of all applicable patterns simultaneously and return them as a list. Some of these prompts use natural language descriptions, while others rely on formalized, rule-based definitions. In contrast, the binary prompts address one pattern at a time and ask the model to decide whether that pattern is present in a yes/no format. Despite their differences, all prompts follow the same format: (1) a Context block that clearly defines the available inputs (e.g., surrounding shots and corresponding transcripts), (2) a Task block that specifies the exact reasoning and output behavior expected of the model, and (3) a list of Patterns, described either in natural language or through rule-based constraints. For multi-class prompts, the expected output is a JSON object of the form:

```
{ "patterns": ["cut-in", "shot-reverse-shot"] }
```

[3] If $i < 3$ or $i > n - 2$, we adjust accordingly (e.g., at the start/end of the video).

To parse these outputs reliably, we implement a two-step fallback strategy. First, we try to directly parse the model's response as JSON and extract values under the key `"patterns"`. If this fails—due to invalid syntax or missing structure—we fall back on regular expression matching to identify known pattern names. If neither approach yields results, we treat the prediction as empty. However, in practice, such failures did not occur, indicating that the prompts successfully constrained model responses. For binary prompts, the model is instructed to return a single value—either 1 (pattern present) or 0 (pattern absent). These outputs are then mapped back to their corresponding patterns, producing prediction vectors compatible with our evaluation framework. All prompt variants used in these experiments are listed in Appendix A.2 and discussed further in Chapter 5.

Formally, let

$$\Phi^{\text{VLM}}_{\text{FEPD}} : W_i \longrightarrow [0,1]^5, \quad \Phi^{\text{VLM}}_{\text{NSD}} : W_i \longrightarrow [0,1]^6.$$

Here, $\Phi^{\text{VLM}}_{\text{FEPD}}(W_i)$ produces a vector indicating film editing pattern $F \in \mathbb{C}_{\text{FEP}}$ being detected within the window W_i while the output vector of $\Phi^{\text{VLM}}_{\text{NS}}(W_i)$ indicates the detection of narrative strategies. For the per-class prompts described above, the prediction system operates as follows. For each film editing pattern $F_J \in \mathbb{C}_{\text{FEP}}$, we define a binary classification function

$$\phi^{\text{VLM}}_{F_j} : W_i \longrightarrow [0,1],$$

where W_i is the five-shot context window from Section 4.3.4, and $\phi^{\text{VLM}}_{F_j}(W_i)$ indicates the probability that F_j occurs in the central span S_i of W_i. We then concatenate the individual per-pattern predictions to form a combined predictor:

$$\widetilde{\Phi}^{\text{VLM}}_{\text{FEPD}}(W_i) = \bigl(\phi^{\text{VLM}}_{F_1}(W_i), \phi^{\text{VLM}}_{F_2}(W_i), \ldots, \phi^{\text{VLM}}_{F_5}(W_i)\bigr),$$

where each $F_j \in \mathbb{C}_{\text{FEP}}$. An analogous approach is used for narrative strategies, resulting in per-strategy functions $\phi^{\text{VLM}}_{NS_j}$ for each $NS_j \in \mathbb{C}_{\text{NS}}$ and a corresponding combined predictor $\widetilde{\Phi}^{\text{VLM}}_{\text{NSD}}(W_i)$.

4.4 Validity Checks

Throughout the pipeline, we conducted several validity checks to ensure correctness. First, we ran some of the processing steps on a small, artificially constructed file with known values, verifying that the code handled edge cases appropriately. We also performed manual reviews of randomly selected segments from actual broadcasts to confirm correct annotation-to-shot alignment and validate feature extractions. Additionally, we pinned the versions of all libraries and models in a requirements file to ensure reproducibility. While these checks do not replace large-scale formal testing, they add a layer of confidence in our methodology and implementation.[4]

[4] For more details on the implementation, see `https://github.com/learebecca/Temporal-Pattern-Exploration-in-News-Videos/`.

5 Experiments

This chapter presents an empirical evaluation of the methods described in Chapter 4, aimed at addressing the three core research questions introduced in Chapter 1. We begin by introducing the annotated dataset, which forms the foundation for all subsequent analyses. An exploratory study follows, aimed at uncovering general trends in the distribution and co-occurrence of film editing patterns and narrative strategies. We then present our experiments on automatic detection of these patterns, evaluating model performance using both standard cross-validation and a cross-domain generalization setup. Finally, we discuss our findings in the context of each research question, note key limitations of the study, and outline potential directions for future work.

5.1 Dataset

This study relies on a dataset of annotated German news videos from five German news outlets. The news outlets were selected to capture a diverse range of editorial styles and journalistic approaches:

- **Tagesschau** (ARD): Germany's most widely watched and trusted public news program, produced by the public broadcaster ARD [81].
- **HeuteJournal** (ZDF): A primetime news program from ZDF, Germany's second major public broadcaster. According to the 2024 Reuters Digital News Report, it is trusted by 62% of Germans [81].
- **Welt** (Axel Springer SE): A private news outlet owned by Axel Springer SE, trusted by 52% of the German population [81].
- **BildTV** (Axel Springer SE): Also produced by Axel Springer SE, BildTV is the video extension of the tabloid-style newspaper *Bild*. It is known for emotionally charged reporting, and *Bild* was the least trusted news source among German respondents in the 2024 Reuters survey [81]. The BildTV format was decommissioned at the end of 2023 [46].
- **COMPACTTV** (Compact-Magazin GmbH): A private video channel of the far-right magazine *Compact*. The channel was temporarily banned due to its right-wing content in July 2024 but reinstated in August 2024. The YouTube channel where it is aired has approximately 500,000 subscribers [25]. Likely due to this comparatively smaller audience, it was not included in the Reuters survey.

All videos are in German, which simplifies the annotation process, linguistic preprocessing, and comparability between channels. However, this language constraint also limits the generalizability of findings to other linguistic and cultural contexts.

5.1.1 Annotation Process

Annotations were performed by a communication scientist at the University of Bremen. Given resource constraints, a single annotator was responsible for each video; thus, no inter-annotator reliability metric is available. This is noted as a limitation in Section 5.7.4. All annotations were produced using the ELAN format [115]. In this format, each pattern occurrence can be tagged with an exact start and end timestamp. A standardized vocabulary was used throughout the annotation process to enable automatic processing of the data.

5.1.2 Dataset Statistics

This section provides general quantitative insights into the annotated dataset. Specifics regarding individual patterns and their distributions are discussed in Section 5.2. In total, the dataset comprises 128 annotated news stories from 65 news videos, amounting to 7.38 hours of video content. On average, each news story lasts approximately 3.46 minutes. The distribution of stories and source videos across the five channels is as follows:

- **Tagesschau**: 51 stories from 23 videos
- **HeuteJournal**: 24 stories from 9 videos
- **Welt**: 20 stories from 18 videos
- **BildTV**: 9 stories from 7 videos
- **COMPACTTV**: 24 stories from 8 videos

This uneven distribution is partly due to methodological considerations. Public broadcasters such as Tagesschau and HeuteJournal exhibited fewer instances of film editing patterns and narrative strategies (Section 5.2). To ensure meaningful representation of patterns across different outlets in the analysis, a greater number of stories were sampled from these outlets.

On a shot level, the dataset contains 3012 shots as detected by the TransNet model. Shot duration was also analyzed, yielding an overall mean shot length of 8.83 seconds (median = 4.24, SD = 15.39). These values suggest a right-skewed distribution with several long-duration outliers. Compared to the historical data reported by Schaefer and Martinez III [99], who observed a trend toward shorter shot durations in television news between 1969 and 2005, our dataset aligns with older styles: our average values fall between those recorded for 1969 (mean = 9.5s, median = 5.3s) and 1983 (mean = 5.2s, median = 3.9s), and are considerably longer than those from 2005 (mean = 4.7s, median = 3.4s). While this might suggest a reversal of the shot-shortening trend, it is more likely due to our exclusive focus on German news formats, which may follow different stylistic conventions.

To better understand differences in editing style, we conducted Mann–Whitney U-tests comparing shot lengths across all channel pairs. The Axel Springer channels (Welt and BildTV)

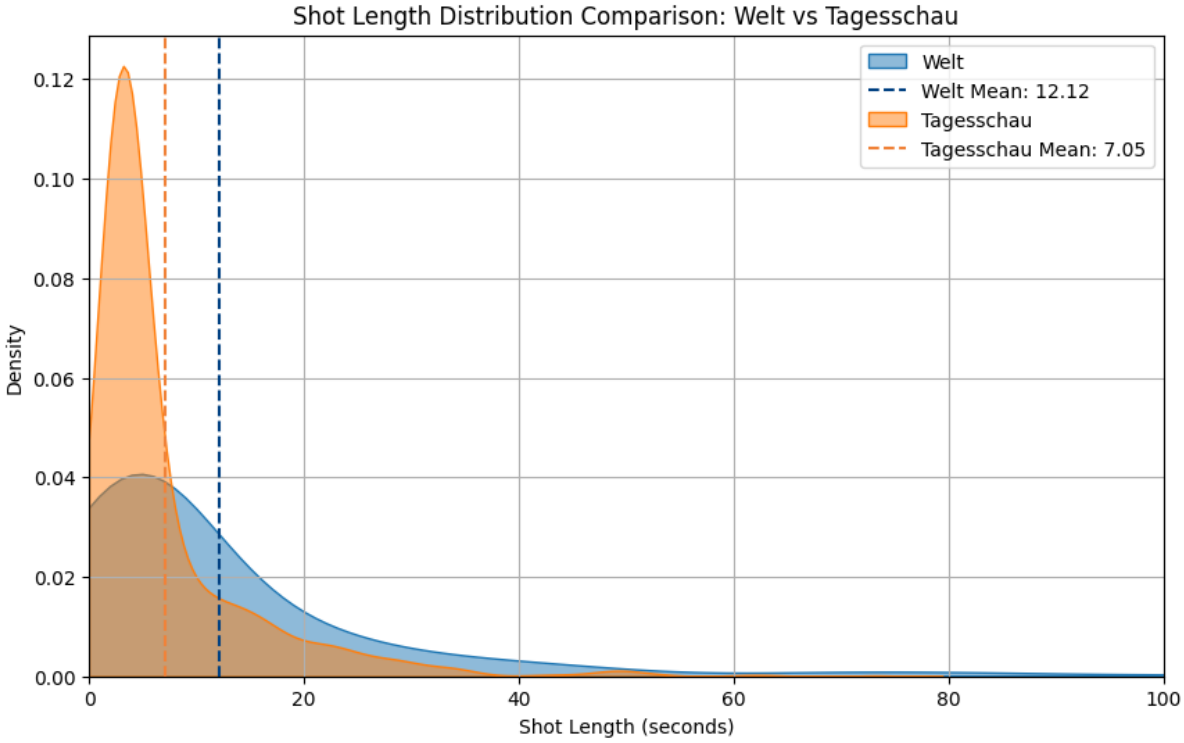


Figure 5.1: Distribution of shot lengths for Tagesschau and Welt

consistently used significantly longer shots than Tagesschau, HeuteJournal, and COMPACTTV[1]. This difference may stem from Axel Springer's frequent use of split-screen editing (Section 5.2), which reduces the need for shot transitions. Such segments are also more challenging to segment with shot detection models like TransNet, as the visual definition of a shot becomes unclear. To illustrate these channel-level differences more clearly, the distributions of shot lengths for Tagesschau and Welt are shown in Figure 5.1.

5.2 Exploratory Analysis of Temporal Patterns

This section explores the occurrence, distribution, and interrelation of film editing patterns (FEPs) and narrative strategies within the annotated dataset. We begin by analyzing the frequency and channel-specific usage of FEPs, including their co-occurrence behavior. We then turn to narrative strategies, examining their distribution across outlets and their coocurence with specific editing patterns. Finally, we investigate the relationships between extracted features—both in terms of inter-feature correlations and their associations with the target patterns. For the explorative analysis, each span was stored in a structured format using `pandas` DataFrames. As described in Chapter 4, fine-grained subcategories—such as alternating-shot variants A through D—were combined into broader categories due to pattern sparsity. Patterns that did not appear in the dataset or were too infrequent to support reliable analysis were excluded from further consideration.

[1] Full results are reported in Appendix A.3

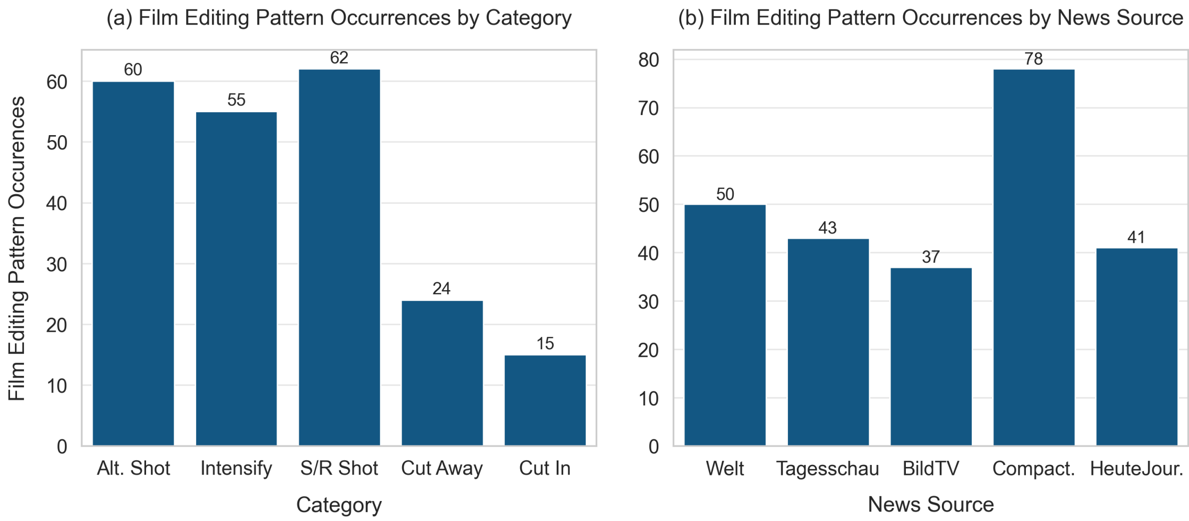


Figure 5.2: Occurrences of film editing patterns in the dataset. (a) shows the distribution by pattern type, and (b) displays the distribution by news source.

5.2.1 Film Editing Patterns

Figure 5.2 (a) illustrates the overall distribution of film editing patterns (FEPs) in the dataset. The most frequently occurring pattern is Shot-Reverse-Shot, followed closely by Alternating-Shot and Intensify. By contrast, Cut-In and Cut-Away occur relatively rarely. Overall, the distribution is sparse, with several theoretically defined patterns not appearing at all in the annotated material.

Table 5.1: Average duration, shot count, and shot duration by pattern type

Pattern	**Avg. Duration (s)**	**Avg. Shots**	**Avg. Shot Duration (s)**
Alternating-Shot	59.83	5.12	11.69
Cut-Away	46.91	4.38	10.72
Shot-Reverse-Shot	36.35	3.70	9.80
Cut-In	13.91	2.73	5.09
Intensify	11.12	2.31	4.82

Table 5.1 presents the average duration of each film editing pattern in seconds, the average number of shots per pattern, and the average shot duration. Alternating-Shot is the longest pattern on average, which is consistent with its definition requiring at least four shots. Cut-Away and Shot-Reverse-Shot also span three or more shots by definition and therefore tend to last longer. Interestingly, Cut-In —which is defined as a three-shot sequence—shows a slightly lower average of 2.73 shots, suggesting that some annotations may deviate from the strict definition, or that shot boundaries were imperfectly detected by TransNet. Intensify, which includes a variant that spans fewer than three shots, has an average of 2.31 shots, indicating the shorter variant is more common.

Furthermore, Alternating-Shot, Shot-Reverse-Shot, and Cut-Away exhibit longer shot durations, which may be attributed to their frequent appearance in extended dialogue sequences. In

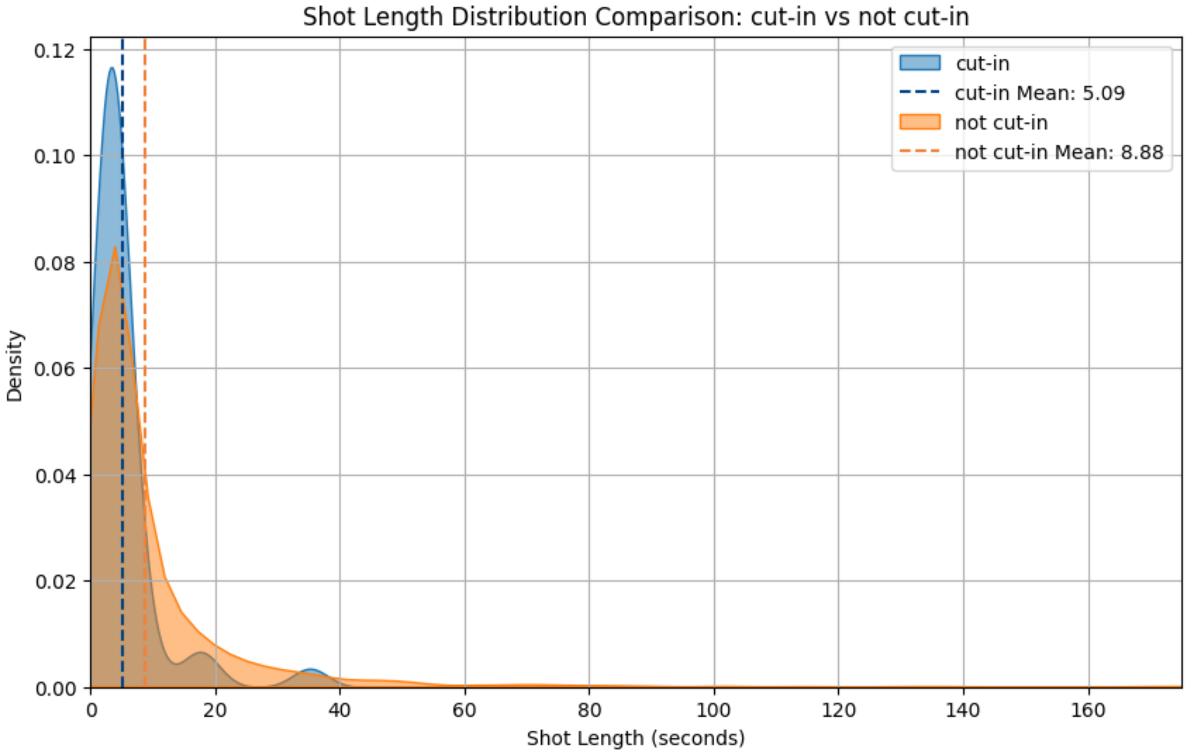


Figure 5.3: Shot length distribution for spans with and without the Cut-In pattern.

contrast, Cut-In and Intensify typically highlight brief moments or specific details and are correspondingly shorter. To determine whether these differences in shot duration are statistically meaningful, we conducted Mann–Whitney U-tests comparing the shot durations within each pattern type to those in all other spans where the respective pattern does not occur. Given the total dataset size of 3,012 shots, the resulting U statistics are relatively large—as expected for comparisons involving groups of this magnitude. The test results are as follows:

- **Alternating-Shot**: $U = 479347.5$, $p < 0.001$, median difference = 0.88 seconds
- **Shot-Reverse-Shot**: $U = 369874.0$, $p < 0.001$, median difference = 2.08 seconds
- **Cut-In**: $U = 48453.0$, $p = 0.024$, median difference = 0.92 seconds
- **Intensify**: $U = 138642.5$, $p < 0.001$, median difference = 1.00 second
- **Cut-Away**: $U = 159374.5$, $p = 0.440$ — not statistically significant

These results support the idea that film editing patterns vary not only in structure but also in pacing, as reflected in their distinct shot length profiles. As an illustration, Figure 5.3 compares the shot length distributions for spans with and without the Cut-In pattern.

Distribution by News Source Figure 5.2 (b) displays the total number of film editing patterns (FEPs) detected per news outlet. While COMPACTTV exhibits the highest raw count of FEPs, this number must be interpreted in the context of the number of annotated stories per outlet. When normalizing by the number of stories, BildTV actually shows the highest density of patterns, with an average of 4.11 FEPs per story, followed by COMPACTTV (3.25), Welt (2.5), HeuteJournal (1.71), and finally Tagesschau (0.84).

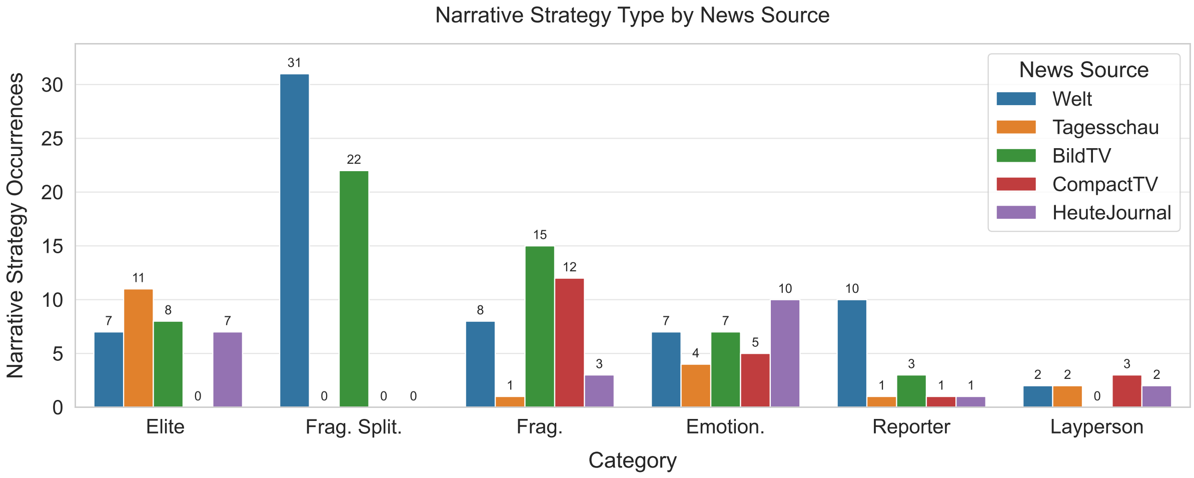


Figure 5.4: Distribution of film editing pattern types across news outlets

These differences are also reflected in the types of patterns used. As shown in Figure 5.4, a key reason for COMPACTTV's high pattern count is its frequent use of the Shot-Reverse-Shot pattern. Tagesschau primarily employs Intensify, while Welt shows a clear emphasis on Alternating-Shot. BildTV frequently uses both Alternating-Shot and Shot-Reverse-Shot, whereas HeuteJournal predominantly uses Shot-Reverse-Shot and Intensify. In general, Intensify appears more often in public broadcasters, while Shot-Reverse-Shot is more commonly used in private outlets. These trends suggest notable differences in editorial style and visual storytelling between public and commercial broadcasters.

Table 5.2: Co-occurrence matrix of film editing patterns. Each cell indicates the number of times a pair of patterns occurred within the same annotated span. Diagonal values represent the total count of each pattern on a span basis.

	alternating-shot	**shot-rev.-shot**	**cut-in**	**intensify**	**cut-away**
alternating-shot	301	14	0	2	57
shot-reverse-shot	14	229	1	0	0
cut-in	0	1	41	3	1
intensify	2	0	3	127	0
cut-away	57	0	1	0	105

Co-occurrence of Patterns Finally, Table 5.2 shows the co-occurrence of different FEPs at the span level. Because this analysis is performed on individual spans rather than entire patterns, the numbers differ from those reported above. Notably, alternating-shot and cut-away overlap frequently (in about half of all cut-away instances, alternating-shot also appears), which is consistent with the idea that alternating-shot involves cutting away and then returning to a previous shot. Shot-reverse-shot and alternating-shot sometimes coincide as well, but overall, FEPs do not co-occur very frequently.

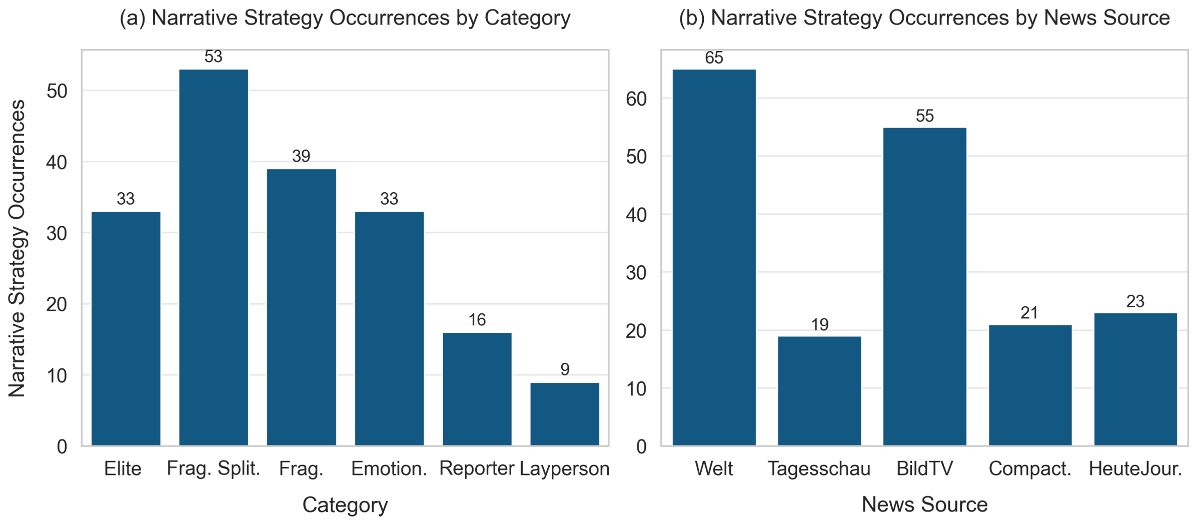


Figure 5.5: Distribution of narrative strategies across all annotated stories. Panel (a) shows total occurrences by strategy category; panel (b) breaks down the usage by news source.

5.2.2 Narrative Strategies

Figure 5.5 (a) shows the overall counts of narrative strategies identified in the dataset. Fragmentation-splitscreen, fragmentation, emotionalization, and individualization of elite occur relatively often, whereas individualization of reporter and individualization of layperson are relatively rare. Dramatization was observed but excluded from further analyses due to their low frequency, which made them unsuitable for inclusion in the training and test sets.

Table 5.3: Average duration, number of shots, and shot length per span for each narrative strategy.

Strategy	Avg. Duration (s)	Avg. Shots	Avg. Shot Duration (s)
reporter	93.18	3.50	26.62
emotionalization	82.85	6.70	12.37
fragmentation-splitscreen	47.97	3.57	13.45
fragmentation	45.18	4.87	9.27
layperson	34.83	5.67	6.15
elite	24.68	2.61	9.47

Table 5.3 reports the average duration, number of shots, and shot duration for each narrative strategy. Overall, these strategies tend to span longer durations than the film editing patterns discussed previously. Notably, individualization of reporter shows a long average shot duration (26.62 seconds). Although this strategy occurs relatively infrequently, a Mann–Whitney U-test confirmed that the difference in shot duration compared to spans without this strategy is statistically significant ($p < 0.001$).[2] The fragmentation-splitscreen strategy deserves special mention. While it is theoretically defined as a single-shot pattern, our setup for the pattern exploration merges directly consecutive instances of the same strategy into one occurence. As

[2] Note that most occurrences of this strategy are found in content from *Welt*, which also has the longest overall shot durations. This may contribute to the elevated average.

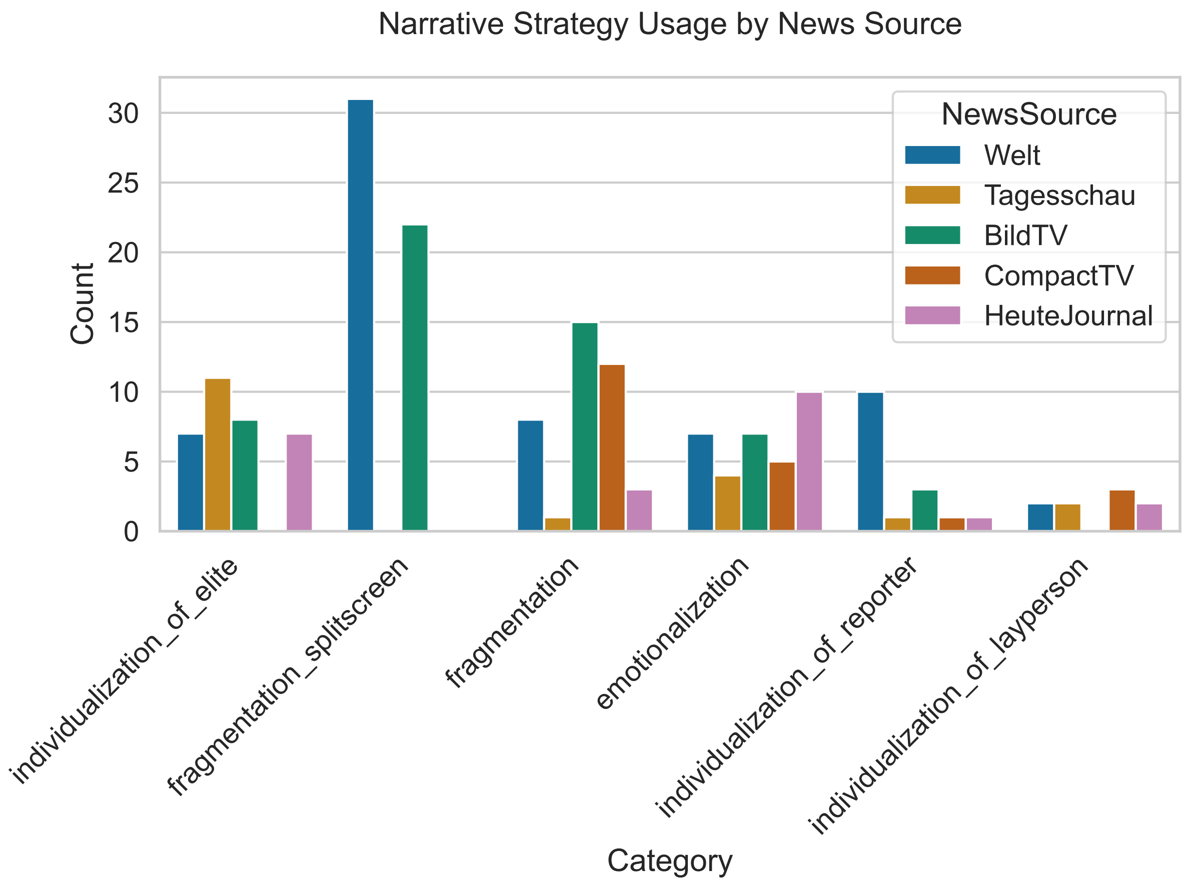


Figure 5.6: Distribution of narrative strategies by news outlet. Each bar reflects the total number of times a given narrative strategy was used by a specific outlet, allowing for comparison of editorial preferences and stylistic emphases.

a result, the reported average duration (47.97s) reflects not the length of a single occurrence but rather the average time over which the pattern appears continuously. When interpreted as a single-shot pattern, the shot duration of 13.45 seconds corresponds directly to its average duration. This value also differed significantly ($p < 0.001$) from the shot durations in spans where fragmentation-splitscreen was not present.

Distribution by News Source Figure 5.5 (b) displays the total number of narrative strategies employed by each news outlet. Welt and BildTV stand out with the highest counts. When normalizing by the number of annotated stories per outlet, they continue to lead, with Welt averaging approximately 1.95 strategies per story and BildTV 1.89. In contrast, the public broadcasters Tagesschau and HeuteJournal make comparatively limited use of narrative strategies, averaging 0.45 and 0.79 strategies per story, respectively. CompactTV also exhibits a lower ratio of 1.04 strategies per story, despite showing the highest frequency of film editing patterns—highlighting a notable difference in stylistic emphasis.

These distinctions become clearer when examining the distribution of individual narrative strategies across outlets in Figure 5.6. Fragmentation-Splitscreen appears exclusively in Welt and BildTV, driving up their overall strategy counts. Emotionalization and Individualization of Layperson are more evenly distributed, while Fragmentation is concentrated in BildTV and CompactTV. Individualization of Elite does not occur in CompactTV, and Reporter individualization is used almost exclusively by Welt, which is also the only outlet to employ every narrative strategy at least once.

Table 5.4: Co-occurrence matrix of narrative strategies based on span-level annotations. Diagonal values represent the total number of spans for each strategy.

	fragment.	elite	reporter	fragment-split.	emotionalization	layperson
fragment.	189	4	0	0	0	0
elite	4	86	0	3	10	0
reporter	0	0	56	19	5	0
fragment-split.	0	3	19	189	26	0
emotionalization	0	10	5	26	221	37
layperson	0	0	0	0	37	51

Co-occurence of Strategies Looking at co-occurrences of strategies in Table 5.4, we observe more overlap than in the case of film editing patterns. Most notably, Layperson co-occurs with Emotionalization in 37 of its 51 instances, suggesting that laypersons are frequently placed in emotionally charged contexts. Fragmentation-Splitscreen also overlaps with Emotionalization in 26 cases. Additionally, about one-third of Individualization of Reporter instances coincide with Fragmentation-Splitscreen, which may reflect the shared prevalence of these strategies in Welt. Notably, Fragmentation and Fragmentation-Splitscreen do not co-occur, potentially because a shot meeting both criteria is coded as Fragmentation-Splitscreen only.

5.2.3 Film Editing Pattern and Narrative Strategy Relationship

Finally, co-occurrence between narrative strategies and film editing patterns shows some noteworthy relationships. Fragmentation-splitscreen often appears together with alternating-shot and shot-reverse-shot. Fragmentation shows the highest overlap with cut-away, and emotionalization co-occurs with alternating-shot and cut-away relatively often. Individualization patterns have fewer co-occurrences, reflecting their overall lower frequency in the dataset.

Table 5.5: Co-occurrence counts between narrative strategies and film editing patterns. Values reflect the number of spans in which both a strategy and a pattern were present.

	alternating-shot	shot-rev.-shot	cut-in	intensify	cut-away
fragmentation	10	1	3	11	17
elite	4	0	0	7	1
reporter	0	0	0	1	0
fragmentation-splitscreen	81	23	0	0	0
emotionalization	63	4	8	4	37
layperson	0	0	10	7	0

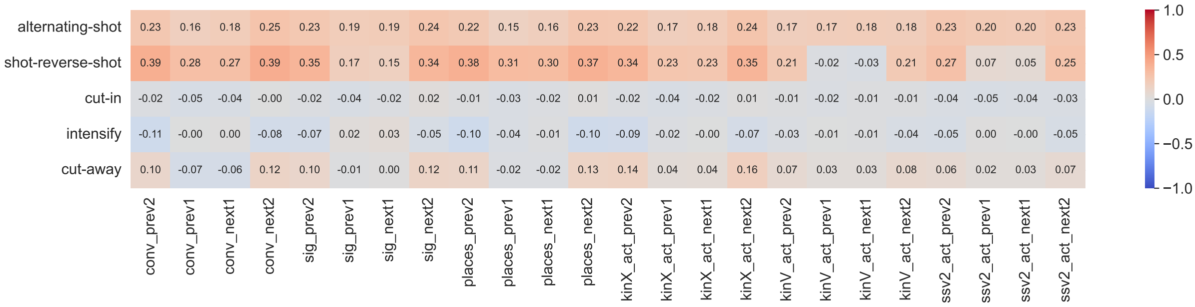


Figure 5.7: Correlation between shot relation features and film editing patterns

5.2.4 Feature Exploration

Before conducting model-based experiments, we explored the extracted features to gain deeper insights into the target variables and assess the relevance of these features for the two prediction tasks: film editing pattern detection (FEPD) and narrative strategy detection (NSD). The goal of this exploration is to better understand (1) how features relate to one another and (2) which features correlate with the target patterns. As noted earlier, the full feature set contains 81 dimensions. These were divided into three subsets: one containing the shot relation features, another containing named entity types and part-of-speech tags, and a third consisting of the remaining features. The features "expert", "layperson", "elite", "talking-head", and "commenting" were excluded from this analysis, as they consistently took the value 0 across all data and were therefore uninformative.

5.2.4.1 Shot Relation Features

Shot similarity and relation features are designed to capture how temporally adjacent shots behave in terms of visual similarity, place or activity similarity. These features capture how similar a shot is to the shot either one before (prev1) two positions before (prev2) or one after (next1) or two after (next2). This is particularly relevant for detecting patterns that rely on temporal editing logic, such as alternating-shot or shot-reverse-shot. For this reason, we expect correlations between these features and the associated patterns. As expected, features computed with respect to the same shot (e.g. conv_next2 and sig_next2 have a correlation of 0.85) are highly correlated. This is unsurprising and adds confidence in the extracted features when different models return similar scores. Conv, sig, and places features show consistently strong correlations (never below 0.75) when calculated for the same reference shot, and kinX_act also shows consistently high correlations (never below 0.66) with these three features. This is noteworthy because kinX_act, while closely related to shot similarity and place similarity, does not correlate as strongly with the other action similarity features. Among the remaining action similarity features, the highest correlation is 0.62. KinV_act and ssv2_act are only moderately correlated with the other features for the same reference shot. Within each similarity feature type, the values for prev2, prev1, next1, and next2 are also moderately related, ranging from a maximum correlation of 0.57 to a minimum of 0.24. No negative correlations were observed. The full correlation heatmap can be found in the Appendix A.4.

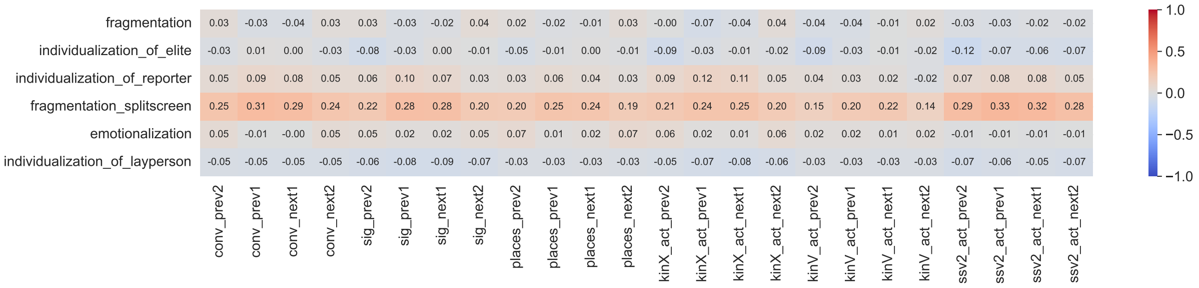


Figure 5.8: Correlation between shot relation features and narrative strategies

Figure 5.7 further explores how shot relation features correlate with FEPs. Most notably, alternating-shot and shot-reverse-shot show moderate correlations with several shot similarity measures. These correlations are consistently stronger for prev2 and next2 than for prev1 and next1, which aligns with the definition of these patterns (s1 and s3 are supposed to be similar to each other as well as s2 and s4). Cut-away is only weakly correlated with the shot relation features, but it also follows the same trend of stronger correlations for prev2 and next2 than for the other two, which also fits its definition. In contrast, intensify tends to exhibit a weak but noticeable negative correlation with prev2 and next2, reflecting the zooming-in behavior, meaning that the shot two positions before is less similar than the one directly before and after. Cut-in does not show any meaningful correlations with these features. Figure Figure 5.8 demonstrates that shot relation features are generally less correlated with narrative strategies. However, fragmentation-splitscreen shows moderate associations with visual similarity. This finding is consistent with earlier observations that this pattern frequently co-occurs with alternating-shot and shot-reverse-shot.

5.2.4.2 Linguistic Features: POS and Named Entities

The linguistic features include part-of-speech (POS) tag frequencies and named entity recognition (NER) counts, all derived from the transcribed speech in each span. Figure Figure 5.9 shows the correlation heatmap between all POS tags and NER categories. POS tags are strongly intercorrelated—particularly common functional tags like ADJ, DET, and NOUN. This can likely be attributed to the aggregation approach, where full speaker turns are always aggregated on a span basis. As a result, we either have no speech at all (and thus no POS or NER occurrences) or we have one or more full speaker turns. Interestingly, POS tags also show moderate correlations with some NER types, especially LOC, ORG, and MISC, but very weak or no correlation with EPER (person names). One exception is PROPN, which correlates more consistently with all NER tags. Proper nouns refer to specific individuals, places, or entities. Therefore, this correlation with NER tags makes sense and again hints at reliable features. The fact that not all POS and NER tags are equally correlated, even with aggregation, suggests that meaningful variation in linguistic structure is still captured despite the aggregation approach.

Figure 5.10 displays the correlation of linguistic features with FEPs. The overall pattern is weak, suggesting that purely linguistic signals may not be very informative for detecting visual editing patterns. Nonetheless, EVENT shows small correlations with alternating-shot and cut-

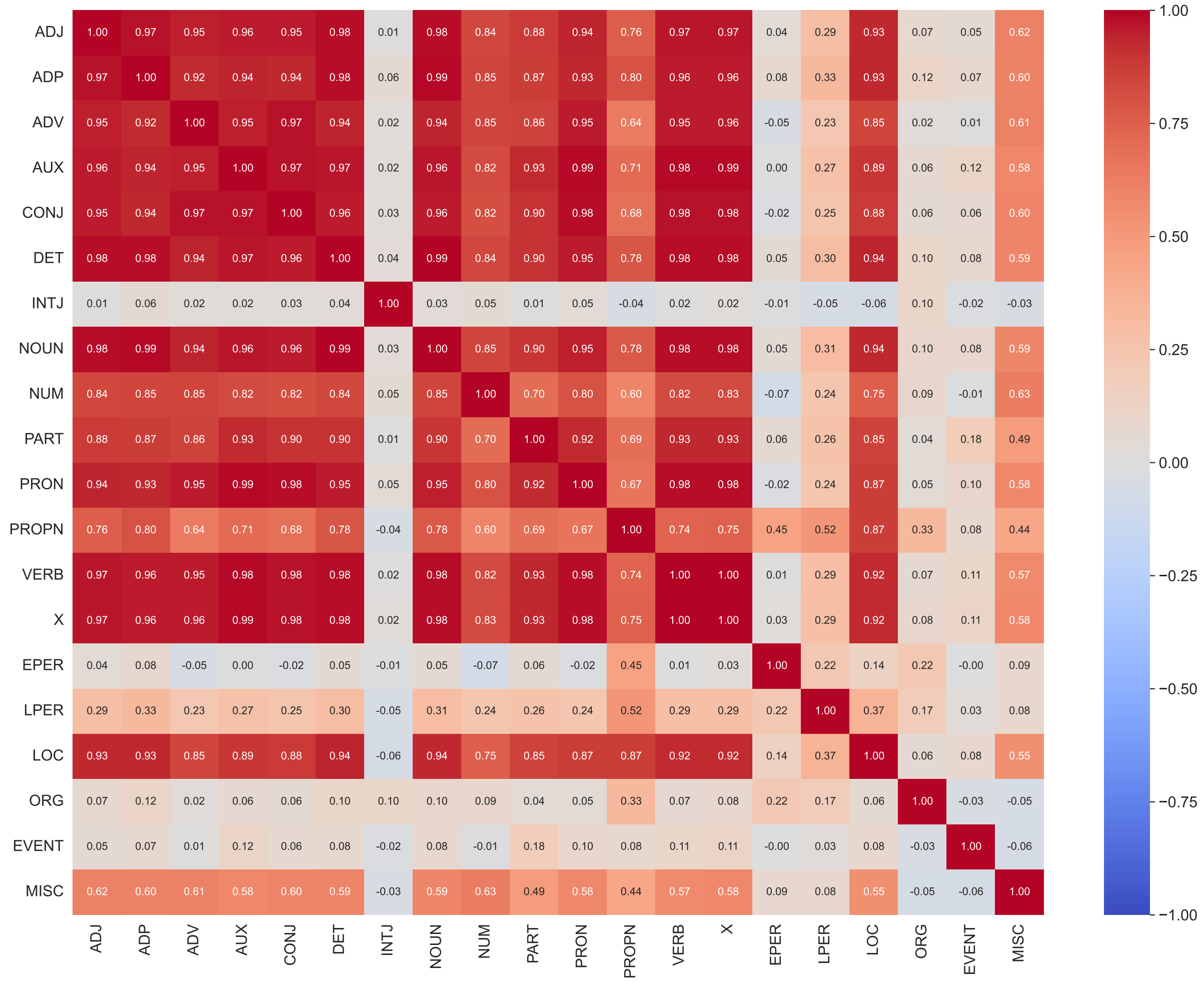


Figure 5.9: Correlation between linguistic features

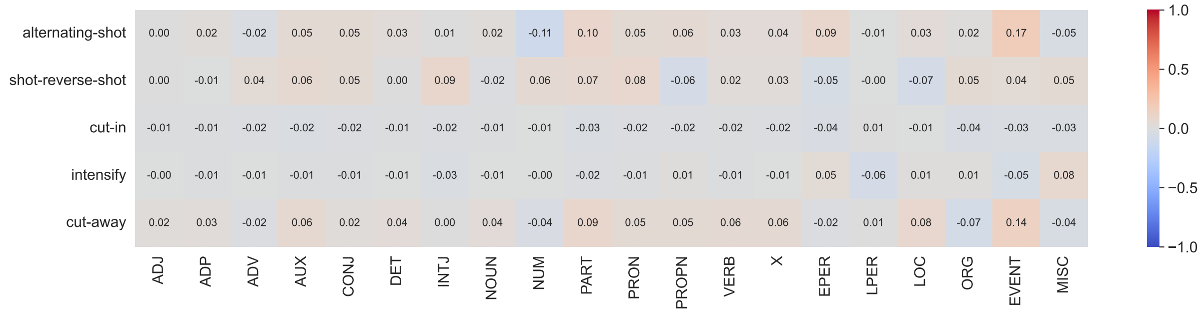


Figure 5.10: Correlation between linguistic features and film editing patterns

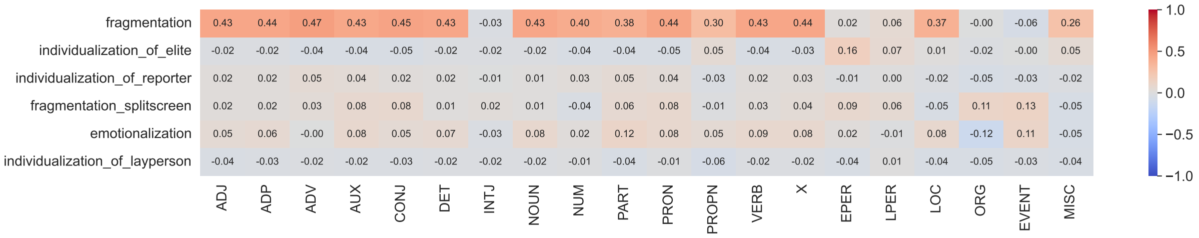


Figure 5.11: Correlation between linguistic features and narrative strategies

away, possibly because such patterns occur while discussing specific events, where related footage is shown. By contrast, Figure 5.11 shows stronger correlations between linguistic features and the strategy fragmentation. Here, correlations are observed across all POS tags (except interjection), as well as with LOC and MISC entities. The presence of correlations with POS tags could indicate that more speech is present during the span. Since fragmentation spans are only slightly longer than average in shot length, this could suggest that longer speaker turns are paired with changing shots, which fits the definition of fragmentation. The correlation with LOC may indicate that location-specific visual material is shown when a location is being discussed. MISC includes all other entity types and is less interpretable. Additionally, individualization of elite shows small but notable correlations with EPER, reinforcing the pattern's focus on prominent individuals.

5.2.4.3 Other Feature Types and General Observations

Additional exploratory analyses were conducted on the remaining numerical features (see Appendix A.4). Generally, these features are not highly correlated with one another. One exception is the pair male_speaker_present and female_speaker_present, which are strongly negatively correlated (-0.85), likely because each span typically includes either only one speaker or gender-consistent interaction patterns. As expected, there are medium negative correlations between features like shot scale and average face size (when we move closer, faces become larger). A notable finding is the absence of a correlation between speech (clip-based) and whisper speech (detected directly from the audio), which may raise questions about the consistency or reliability of these features.

We also analyzed the correlation between features and the target variables. These correlations were generally weaker. For brevity, we highlight only specific target types. For shot-reverse-shot, we observe a moderate positive correlation with interview and a moderate negative correlation with voice-over. There are also smaller correlations with angry, fear, llm_evaluative, face size, and face region (suggesting a preference for lower-right framing). Shot-reverse-shot is negatively correlated with shot density and videoshot_scalemovement. Due to the encoding of videoshot_scalemovement, this indicates that the pattern tends to occur when shots are closer, which supports the interpretation of shot-reverse-shot as a pattern often used in conversations, particularly interviews that may be emotional in tone. As with linguistic and shot relationship features, Cut-in does not show any meaningful correlations with these available features. This might make it difficult to detect the pattern on this basis. Finally, regarding narrative strategies, it is surprising to find a lack of correlation between the emotion features (deepface) and the strategy emotionalization. One possible explanation is the aggregation strategy applied at the span level, which may dilute momentary expressions of emotion as discussed in Section 4.3.1.

These findings helped inform which features to prioritize in downstream experiments. In particular, the strong correlation of visual similarity features with certain FEPs supported their inclusion as core inputs.

5.3 Experimental Setup for Pattern Detection

This section outlines the experimental setup used to evaluate the two central prediction tasks in this thesis: Film Editing Pattern Detection (FEPD) and Narrative Strategy Detection (NSD). The goal of these experiments is to assess how well machine learning models can identify recurring stylistic and narrative patterns in video news content, based on the feature representations introduced in Chapter 4.2.2. To this end, we define consistent protocols for data splitting, evaluation metrics, baseline comparisons, model training, hyperparameter tuning, and feature selection. These choices are informed by both practical considerations—such as class imbalance and span-level annotation—and by theoretical motivations, including generalizability. The subsections below describe each of these components in detail.

5.3.1 Data Splits

This evaluation uses two main setups: a five-fold configuration and a cross-domain configuration. In the five-fold cross-validation setup, 20% of the dataset is first held out as a final test set. The remaining data is split into five folds for training and validation. To address the substantial class imbalance, the split is not performed completely at random. Random splitting risks omitting rare classes from validation or test sets entirely. Instead, the splitting process begins with the rarest class, distributing stories that contain it evenly across the five folds. This process is repeated for the second-rarest class, and so on, before finally distributing stories that contain no patterns. Since splitting is performed at the story level, class balance is not guaranteed but is significantly improved compared to random assignment. Due to the low frequency and non-overlapping distributions of some labels, it was not feasible to use the exact same splits for both FEPD and NSD. Therefore, distinct folds were created for each task. When evaluating narrative strategies, the strategy-specific split is used; for film editing patterns, the pattern-specific split is applied.

In the cross-domain setup, models are trained on data from four of the five available source channels and tested on the remaining one. This configuration is intended to evaluate generalization across different stylistic and production conditions—a scenario more aligned with real-world deployment. Based on exploratory data analysis, Welt was selected as the held-out test domain, as it is the only channel that includes all defined film editing patterns and narrative strategies.

5.3.2 Evaluation Metrics

Selecting an appropriate evaluation metric for temporal pattern detection can be challenging, as many existing approaches rely on exact or approximate temporal boundary matching (e.g., fuzzy precision/recall or window-based metrics) [**hauptmann1997informedia**, 59, 93]. However, in this work, all annotations are provided at the shot level. This means that if a shot is classified correctly, the corresponding temporal boundaries are also inherently correct. Therefore, we frame this as a multi-label classification problem and adopt multi-label classification metrics. Given the multi-label nature of the task, metrics are computed per label and aggregated using micro-averaging. This approach aggregates counts across all labels before calculating the final metric, ensuring that each label prediction contributes proportionally to the result. While macro-averaged metrics may become more important in future work, especially to address rare class performance, our current focus is on assessing general model capability in a highly imbalanced setting with limited annotations for some classes. We report micro-averaged precision, recall, F1, and F2 scores. Emphasis is placed on the F2 score, which weights recall twice as heavily as precision. This is particularly relevant in our setting: since most shots contain no patterns, a model can trivially achieve high accuracy by predicting the absence of all patterns. However, our goal is not to replace human annotators but to assist them by identifying potentially relevant segments—making recall more valuable than precision at this stage. For completeness, we also report subset accuracy and Hamming loss, although these are less informative under heavy class imbalance. All metrics are implemented using Pedregosa et al. [88].

Let TP be the true positives (correctly classified patterns across all labels), FP the false positives (instances incorrectly classified as containing the pattern), and FN the false negatives (missed patterns across all labels).

Micro-Precision:

$$\text{Micro-Precision} = \frac{TP}{TP + FP}$$

Micro-Recall:

$$\text{Micro-Recall} = \frac{TP}{TP + FN}$$

Micro-F1:

$$\text{Micro-F1} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Micro-F2:

$$\text{Micro-F2} = \frac{5 \cdot TP}{5 \cdot TP + FP + 4 \cdot FN}$$

Subset Accuracy: This metric measures the proportion of instances for which all predicted labels exactly match the true labels. Let N be the number of instances, and $\hat{y}_i$ and y_i the predicted and true label sets for instance i. Subset accuracy requires all labels for an instance to match exactly:

$$\text{Subset Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i)$$

Hamming Loss: For multi-label tasks, this quantifies the fraction of incorrect label predictions over all labels and instances:

$$\text{Hamming Loss} = \frac{1}{N \cdot L} \sum_{i=1}^{N} \sum_{j=1}^{L} \mathbb{I}(y_{ij} \neq \hat{y}_{ij})$$

where L is the total number of labels, and $y_{ij}, \hat{y}_{ij} \in \{0, 1\}$ are the ground truth and predicted binary values for label j of instance i.

All classifiers were trained using three random seeds, and metrics were averaged across these trials.

5.3.3 Baselines

To contextualize model performance, we evaluate three simple heuristic baselines:

- **Always-Negative Baseline:** Predicts no pattern for any instance.
- **Random Baseline:** Predicts each pattern with 50% probability.
- **Always-Positive Baseline:** Predicts every pattern as present in every instance.

These baselines provide reference points for performance and help gauge whether classifiers are learning meaningful distinctions or simply benefiting from class imbalance.

In addition to these heuristics, we include a vision-language model (VLM) as an external reference point. The system tested is Qwen2.5-VL-3B-Instruct [10], evaluated in both per-class and multi-label settings. In the multi-label setup, two prompting strategies are used: one based on formal pattern definitions, and another using natural language descriptions aligned with NLP conventions. The per-class configuration combines elements of both. These prompts were optimized using the chat version Qwen2.5 during development and thus differ slightly from the formal definitions presented in this thesis. General prompting methodology is discussed in Chapter 4, and all prompt templates are included in Appendix A.2. While the VLM was not the primary focus of this work, its inclusion serves to benchmark the zero-shot performance of large multimodal models relative to more targeted feature-based classifiers.

In addition to these heuristics, we include a vision-language model (VLM) as an external benchmark. The system tested is *Qwen2.5-VL-3B-Instruct* [10], evaluated in both per-class and multi-label settings. In the multi-label setup, two prompting strategies are used: one based on formal pattern definitions, and another using natural language descriptions aligned with NLP conventions. The per-class configuration combines elements of both. These prompts were optimized using the chat version *Qwen2.5* and thus differ slightly from the formal definitions presented in this thesis. General prompting methodology is discussed in Chapter 4, and all prompt templates are provided in Appendix A.2. While the VLM was not the primary focus of this work, its inclusion helps benchmark the zero-shot performance of large multimodal models relative to the more targeted feature-based classifiers.

5.3.4 Classifiers

Two model types are trained for the classification tasks described in **??**: Random Forest and Gradient Boosting. These models are chosen for their robustness and comparatively straightforward interpretability with regards to feature importance, which is crucial for applications in semi-automated news content analysis and for researchers in media studies and journalism who need clear explanations of model outputs. The Random Forest (RF) classifier and the Gradient Boosting classifier (GBC) are both implemented using the `scikit-learn` library [88] in Python. Both model types have been successfully applied in related work Cheema et al. [21], further supporting their relevance to this task.

5.3.5 Hyperparameter Tuning

For the Random Forest classifier, we tune the number of estimators, the maximum tree depth, the maximum number of features considered at each split, and the minimum number of samples required to remain after a split. All combinations of these values are evaluated using a grid search. Preliminary experiments showed that the number of estimators had the greatest impact on performance. Therefore, we test five, 100, 300, and 500 estimators. For maximum depth, we compare the default value (`None`) with a restricted depth of five. For the maximum number of features, we evaluate using all features, the square root of the total number, and half of the features. The minimum number of samples per split is varied between the default value (two) and a more restrictive setting (eight), based on early results showing that values near two had minimal effect. For Gradient Boosting, we tune the same values for the number of estimators, maximum depth, and maximum number of features. Instead of the minimum samples per split, we tune the learning rate. Specifically, we test a small rate of 0.01, the `scikit-learn` default of 0.1, and the Gradient Boosting Library default of 0.3 [**xgb**]. We fix the subsample ratio at 0.8, based on initial tests showing slightly better performance than the default value of one.

This setup results in 48 parameter configurations for Random Forest and 72 for the Gradient Boosting Classifiers. Each configuration is evaluated for both multi-label and per-class settings, averaged over three random seeds and five-fold cross-validation, yielding a large number of trained models. Specifically, 48 configurations × five folds × three seeds yields 720 Random Forest models per task in the multi-label setup. In the per-class setting, an additional 3,600 RF models are trained for narrative strategies, and 4,320 for film editing patterns, totaling 9,360 Random Forest models overall. For Gradient Boosting Classifiers, 72 configurations lead to 14,040 models across both task setups. This results in a combined total of 19,080 trained models across both classifiers. The best-performing configuration for each task is selected based on the highest F2 score on the validation set. These final configurations—one each for multi-label and per-class settings for both tasks—are reported in the subsequent experiments section.

5.3.6 Feature Sets

An 81-dimensional feature vector is extracted for each span, as detailed in Chapter 4.2.2. While this dimensionality is modest compared to raw video or audio data, it still represents a heterogeneous feature space composed of various modalities. Rather than applying automated feature selection prior to training, we designed a series of controlled analyses to explore the influence of different feature groupings. These analyses are conducted separately for each prediction task but are based on consistent feature subsets. Below, we describe the six sets used throughout.

Vision: This feature set includes all features derived solely from visual data, resulting in a total of 37 features. This includes, for example, shot relation features, shotscale, and face emotions.

Audio: This set consists of all features based on audio data. It also includes features such as named entity recognition, which were extracted from the textual transcript. In total, this set comprises 33 features.

Multimodal: This set contains features that integrate both visual and audio information. It includes 12 features: active speaker, unique speakers, anchor, reporter, expert, layperson, elite, interview, talking-head, speech, commenting, and voice-over. However, the features expert, layperson, elite, talking-head, and commenting consistently take the value 0 across the dataset. As a result, only 7 features in this group are actively used by the models.

Audio and Multimodal: To provide a comparison set equal in size to the visual feature set, this set combines all audio features with a selected subset of the multimodal features: anchor, reporter, interview, speech, and voice-over. The feature female speaker present was removed due to its strong negative correlation with male speaker present. This results in a total of 37 features, matching the visual set in size.

Highest Correlations: We also selected the 37 features with the highest absolute correlation with any of the target variables. All features with an absolute correlation above 0.245 were included. The resulting set comprises the following: ADJ, ADP, ADV, AUX, CONJ, DET, LOC, MISC, NOUN, NUM, PART, PRON, PROPN, VERB, X, anchor, conv_next1, conv_next2, conv_prev1, conv_prev2, interview, kinX_act_next1, kinX_act_next2, kinX-act_prev2, places_next1, places_next2, places_prev1, places_prev2, sig_next1, sig_next2, sig_prev1, sig_prev2, ssv2_act_next1, ssv2_act_next2, ssv2_act_prev1, ssv2_act_prev2, and voice-over.

Handcrafted Set: Lastly, we curated a smaller, handcrafted feature set designed to minimize redundancy while ensuring diversity across modalities. It includes 28 features: convnextv2-shotsimilarity features, videoshot scalemovement, ssv2-vmae-action-shotsimilarity features, llm evaluative, active speaker, anchor, reporter, interview, speech, voice-over, region_y, angry, fear, happy, Speech, eper, org, event, adv, part, and pron.

These feature sets provide the foundation for a series of experiments that investigate both overall classification performance and the comparative value of different modalities. In the next sections, we present the results for each task.

5.4 Experiments for Film Editing Pattern Detection

This section presents the results for the task of Film Editing Pattern Detection (FEPD). We evaluate two types of classifiers—Random Forest (RF) and Gradient Boosting Classifier (GBC)—in both multi-class and per-class configurations. In addition to these models, we include a vision-language model (VLM) and several heuristic baselines to provide a broader performance context. We structure the evaluation into three parts: first, we assess overall model performance using five-fold cross-validation; second, we test model generalizability under domain shift using a cross-channel setup; and third, we analyze the impact of different feature subsets on detection performance. Together, these experiments provide a comprehensive picture of classifier behavior and feature importance for FEPD, and they align with the experimental protocols introduced in Chapter **??**.

The following hyperparameter configurations yielded the highest validation F2 scores during tuning and are used throughout this section:

- **XGB Multi-label:** five estimators, max depth five, no feature limit, learning rate 0.3.
- **RF Multi-label:** five estimators, no max depth, max features 0.5, minimum samples split eight.
- **XGB Per-class:** five estimators, max depth five, max features `sqrt`, learning rate 0.01.
- **RF Per-class:** 500 estimators, max depth five, no feature limit, minimum samples split two.

5.4.1 Evaluation under K-Fold Cross-Validation

Table 5.6 presents the results for the FEPD task using the five-fold cross-validation split. The best overall performance, as measured by the F2 score, is achieved by the per-class Random Forest model (**F2 = 0.4494**), followed closely by the per-class Gradient Boosting Classifier (**F2 = 0.4335**). These two models generally outperform their multi-label counterparts, particularly in terms of recall, which is critical for achieving higher F1 and F2 scores. Interestingly, the multi-label Random Forest model yields the highest precision (**0.3339**) and—excluding the always-negative baseline—the best subset accuracy (**0.7583**) and the lowest Hamming Loss (**0.0537**). This suggests that the model is conservative in its predictions, making fewer false positives. However, this conservative behavior comes at the cost of lower recall, indicating a tendency to underpredict the presence of editing patterns.

Table 5.6: Evaluation results for the FEPD task using the k-fold split.

Model Name	**Precision**	**Recall**	**F1**	**F2**	**SubsetA.**	**H. Loss**
GBC-multilabel	0.1867	0.0807	0.1126	0.0910	0.7262	0.0623
GBC-perclass	0.2331	**0.5528**	0.3277	0.4335	0.5252	0.1111
RandomForest-multilabel	**0.3339**	0.0994	0.1530	0.1156	**0.7583**	**0.0537**
RandomForest-perclass	0.2627	0.5466	**0.3548**	**0.4494**	0.5735	0.0977
VLM:Descriptions	0.0154	0.0035	0.0058	0.0042	0.7105	0.0613
VLM:Definitions	0.0132	0.0035	0.0056	0.0042	0.7007	0.0632
VLM:Per-class	0.0551	0.0248	0.0342	0.0279	0.7114	0.0702
Baseline-All	0.0501	1.0000	0.0954	0.2086	0.0000	0.9499
Baseline-None	0.0000	0.0000	0.0000	0.0000	0.7593	0.0501
Baseline-Random	0.0452	0.4504	0.0822	0.1614	0.0302	0.5036

The vision-language model (VLM) variants perform notably worse than the tree-based classifiers, with all configurations yielding F2 scores below 0.03. Among them, the per-class VLM prompt performs best but still trails behind even the random baseline in terms of F2. This underscores the limitations of zero-shot pattern recognition in this setting. As expected, the always-negative baseline achieves the highest subset accuracy (**0.7593**) and lowest Hamming Loss (**0.0501**) by simply predicting the majority class (no pattern). However, it fails entirely to identify any patterns. The always-positive baseline, by contrast, achieves perfect recall but suffers from extremely low precision and high error rates.

5.4.2 Evaluation under Cross-Domain Setup

Table 5.7 shows model performance on the generalization split, in which classifiers are trained on four channels and evaluated on a held-out channel (Welt). As expected, overall performance decreases compared to the k-fold setting due to the domain shift. Nevertheless, the per-class Random Forest remains the strongest model, achieving the highest F2 score (**0.1977**), as well as the best results in terms of precision (**0.1834**), recall (**0.2018**), and F1 score (**0.1921**). While the absolute values drop across all metrics, this model shows greater robustness to out-of-distribution data and generalizes more effectively than other classifiers. Gradient Boosting Classifiers' per-class configuration also performs relatively well in this setting (**F2 = 0.1805**), confirming its comparative strength over multi-label variants. In contrast, the multi-label models—particularly Random Forest—perform substantially worse in terms of recall and F-scores, though the RF multi-label variant achieves the highest subset accuracy (**0.7274**) and lowest Hamming Loss (**0.0726**). This pattern mirrors the trend seen in the k-fold evaluation, where the multi-label models act more conservatively but fail to detect most patterns.

Table 5.7: Evaluation results for the FEPD task using the generalization split.

Model Name	**Precision**	**Recall**	**F1**	**F2**	**Subset A.**	**H. Loss**
GBC-multilabel	0.0567	0.0140	0.0225	0.0165	0.7090	0.0779
GBC-perclass	0.1412	0.1947	0.1631	0.1805	0.5268	0.1250
RandomForest-multilabel	0.0475	0.0088	0.0148	0.0105	**0.7274**	**0.0726**
RandomForest-perclass	**0.1834**	**0.2018**	**0.1921**	**0.1977**	0.5897	0.1077
VLM:Descriptions	0.1500	0.0474	0.0720	0.0549	0.6873	0.0776
VLM:Definitions	0.1746	0.0579	0.0870	0.0668	0.6856	0.0773
VLM:Per-class	0.0536	0.0316	0.0397	0.0344	0.6940	0.0970
Baseline-All	0.0635	1.0000	0.1195	0.2533	0.0000	0.9365
Baseline-None	0.0000	0.0000	0.0000	0.0000	0.7625	0.0635
Baseline-Random	0.0626	0.5000	0.1113	0.2086	0.0301	0.5074

VLM performance improves slightly compared to the k-fold evaluation. The definitions-based prompt yields the second-highest precision among all models (**0.1746**), but recall remains low (**0.0579**), resulting in a modest F2 score (**0.0668**). The per-class VLM configuration again fails to outperform even the random baseline across recall-oriented metrics, reinforcing the observation that the selected vision-language model struggles with this task in a zero-shot setting.

The heuristic baselines continue to highlight the effects of class imbalance. The always-negative baseline again achieves the highest subset accuracy (**0.7625**) and lowest Hamming

Loss (**0.0635**) simply by predicting the absence of all patterns. In contrast, the always-positive baseline attains perfect recall but extremely low precision and no correct label combinations. Interestingly, the random baseline achieves higher F1 and F2 scores than both the VLMs and some classifier variants in this generalization setting. However, this comes at the cost of very poor subset accuracy (**0.0301**) and a high Hamming Loss (**0.5074**), underlining its lack of discriminative ability.

5.4.3 Evaluation of Feature Sets for Film Editing Pattern Detection

To address RQ2 — identifying which types of features are most informative for detecting film editing patterns — we evaluate model performance across a range of feature subsets. These subsets, defined in Section 5.3.6, include unimodal groups (e.g., visual or audio features only), multimodal features, handcrafted selections, and correlation-based rankings. Each subset is tested using both Gradient Boosting classifiers and Random Forest classifiers in a per-class setting, applying the best-performing hyperparameter configurations identified for the Film Editing Pattern Detection (FEPD) task. While the multi-label configuration was also evaluated, we focus here on the per-class results due to the volume of tests conducted. The general trends were similar in both settings.

Table 5.8 summarizes the results for each feature subset under the k-fold setup. Across both classifiers, the visual feature set consistently delivers the strongest performance, with a Gradient Boosting classifier achieving an F2 score of **0.4571** and Random Forest reaching **0.4979**. These results are largely driven by the high recall values (**0.6190** and **0.6791**, respectively), with Random Forest showing the highest recall of any configuration.

Table 5.8: Feature experiment results for FEPD on k-fold split.

Model	**Feature Set**	**Precision**	**Recall**	**F1**	**F2**	**SubsetA.**	**H.Loss**
GBC	all	0.2331	0.5528	0.3277	0.4335	0.5252	0.1111
GBC	audio	0.0879	0.2443	0.1285	0.1787	0.4087	0.1638
GBC	visual	0.2238	0.6190	0.3285	0.4571	0.4789	0.1245
GBC	multimodal	0.1057	0.6501	0.1819	0.3202	0.1267	0.2877
GBC	audio&multi	0.0848	0.2588	0.1274	0.1829	0.3471	0.1766
GBC	correlations	0.1723	0.4493	0.2491	0.3400	0.4280	0.1332
GBC	handcrafted	0.1956	0.4638	0.2745	0.3630	0.4774	0.1210
RandomF.	all	**0.2627**	0.5466	0.3548	0.4494	**0.5735**	**0.0977**
RandomF.	audio	0.1196	0.2257	0.1563	0.1916	0.5226	0.1195
RandomF.	visual	0.2409	**0.6791**	**0.3556**	**0.4979**	0.4896	0.1210
RandomF.	multimodal	0.1071	0.6398	0.1835	0.3207	0.1405	0.2799
RandomF.	audio&multi	0.1509	0.3251	0.2060	0.2640	0.4870	0.1231
RandomF.	correlations	0.1865	0.4638	0.2660	0.3574	0.4545	0.1258
RandomF.	handcrafted	0.2414	0.5424	0.3341	0.4341	0.5226	0.1064

Interestingly, despite its small size, the multimodal feature set also achieves relatively high recall scores—**0.6501** for Gradient Boosting Classifier and **0.6398** for Random Forest—suggesting that a small number of carefully designed multimodal indicators may still carry crucial information. However, when this set is combined with the full audio feature set (in the audio&multi

configuration), performance drops across most metrics. This indicates that the additional information may introduce noise rather than improve predictions. The handcrafted set also shows performance equal to that of including all features, especially when combined with the random forest model. This further underscores the trend of quality over quantity features.

In contrast, the audio-only and audio&multi subsets perform the worst across nearly all metrics, highlighting the limited standalone value of non-visual signals for this task. While these features may still offer complementary cues when fused with visual input, they appear insufficient on their own. Notably, the full feature set yields the highest precision for both models, suggesting that larger, more diverse feature combinations help reduce false positives, even if they do not always maximize recall.

Table 5.9 reports the same feature subset comparisons on the generalization split, where models are evaluated on unseen data from the held-out channel (Welt). As expected, performance drops across the board due to domain shift. However, the visual feature set remains the most robust in this setting as well, particularly with Random Forest (**F2** = **0.2883**). The performance decline for the visual subset is less severe than for the full feature set, indicating better generalization. Although multimodal subsets still yield relatively high recall, their precision remains low, resulting in weaker F2 scores overall. Audio-only and audio-enhanced subsets continue to underperform, especially in generalization, which confirms their limited utility without strong visual signals.

Table 5.9: Feature experiment results for FEPD on the cross-domain setup.

Model	Featureset	Precision	Recall	F1	F2	SubsetA.	H.Loss
GBC	all	0.1412	0.1947	0.1631	0.1805	0.5268	0.1250
GBC	audio	0.0836	0.1719	0.1117	0.1409	0.3807	0.1759
GBC	visual	0.1633	0.3123	0.2143	0.2638	0.4292	0.1457
GBC	multimodal	0.0809	0.3228	0.1293	0.2017	0.1773	0.2784
GBC	audio&multi	0.0702	0.1579	0.0968	0.1258	0.3645	0.1834
GBC	correlations	0.0908	0.1737	0.1190	0.1466	0.4114	0.1620
GBC	handcrafted	0.1301	0.2123	0.1611	0.1883	0.4509	0.1392
RandomF.	all	**0.1834**	0.2018	0.1921	0.1977	**0.5897**	**0.1077**
RandomF.	audio	0.0645	0.0965	0.0772	0.0876	0.4509	0.1460
RandomF.	visual	0.1773	**0.3421**	**0.2334**	**0.2883**	0.4298	0.1423
RandomF.	multimodal	0.0823	0.3228	0.1311	0.2037	0.2441	0.2719
RandomF.	audio&multi	0.0575	0.0895	0.0700	0.0805	0.4565	0.1505
RandomF.	correlations	0.0874	0.1193	0.1008	0.1111	0.4855	0.1346
RandomF.	handcrafted	0.1037	0.1421	0.1198	0.1322	0.4721	0.1326

Taken together, these findings strongly support the central role of visual features in film editing pattern detection and suggest that compact, well-designed subsets may offer a good balance between performance and interpretability in practical applications as well as the potential for better generalization.

5.5 Experiments for Narrative Strategy Detection

This section presents the results for the task of Narrative Strategy Detection (NSD). As in the Film Editing Pattern Detection experiments, we evaluate both classifiers in multi-class and per-class configurations. In addition to these models, we include a vision-language model (VLM) and several heuristic baselines to provide broader performance context. Results are reported for both the five-fold cross-validation and cross-domain generalization setups, as described in **??**.

For each model type, the best-performing hyperparameter configuration was selected based on the highest F2 score on the validation set for NSD. These configurations are applied consistently across all evaluation scenarios presented in this section:

- **XGB Multi-label:** five estimators, max depth five, max features 0.5, learning rate 0.3
- **RF Multi-label:** five estimators, no max depth, max features 0.5, minimum samples split eight
- **XGB Per-class:** 100 estimators, no max depth, all features, learning rate 0.01
- **RF Per-class:** 100 estimators, max depth five, max features 0.5, minimum samples split eight

5.5.1 Evaluation under K-Fold Cross-Validation

Table 5.10 presents the evaluation results for the Narrative Strategy Detection (NSD) task under the five-fold cross-validation setup. Among the model configurations, the per-class Gradient Boosting model achieves the best overall performance, with the highest F2 score (**0.2989**) and F1 score (**0.2709**). It also attains the highest recall (**0.3211**), indicating a stronger ability to identify narrative strategies compared to the other variants. In general, however, recall scores for narrative strategy detection tend to be lower than those observed in the film editing pattern detection task, suggesting that strategies may be more difficult to capture automatically. The per-class Random Forest model performs competitively, though slightly behind Gradient Boosting in both recall and F2.

Notably, the multi-class Random Forest model achieves the highest precision (**0.4952**), the best subset accuracy (**0.7331**), and the lowest Hamming Loss (**0.0535**). However, its limited recall (**0.1374**) restricts its effectiveness in recall-focused metrics like F2, again reflecting a more conservative prediction strategy—consistent with the trends observed in FEPD. The multi-class Gradient Boosting configuration underperforms its per-class counterpart across all metrics, particularly recall and F2, suggesting that the per-class formulation is better suited to capturing the structural and semantic nuances of narrative strategies.

As with the FEPD task, the visual language model baselines perform poorly across all metrics. The best-performing VLM configuration (per-class prompting) achieves an F2 score of only **0.1243**, well below both tree-based models and even the random baseline. The always-positive heuristic baseline achieves perfect recall, but at the cost of very low precision and a high Hamming Loss, while the always-negative baseline achieves the highest subset accuracy (**0.7180**) by predicting the absence of all strategies.

Table 5.10: Evaluation results for NSD task using the k-fold split.

Model Name	Precision	Recall	F1	F2	SubsetA.	H.Loss
GBC-multilabel	0.2363	0.1211	0.1592	0.1338	0.6758	0.0670
GBC-perclass	0.2343	**0.3211**	**0.2709**	**0.2989**	0.5742	0.0919
RandomForest-multilabel	**0.4952**	0.1374	0.2145	0.1605	**0.7331**	**0.0535**
RandomForest-perclass	0.2428	0.2190	0.2302	0.2234	0.6445	0.0778
VLM:Description	0.0400	0.0202	0.0268	0.0224	0.6249	0.0820
VLM:Definition	0.0410	0.0327	0.0364	0.0341	0.5495	0.0971
VLM:PerClass	0.0633	0.1637	0.0913	0.1243	0.4462	0.1826
Baseline-All	0.0560	1.0000	0.1061	0.2288	0.0000	0.9440
Baseline-None	0.0000	0.0000	0.0000	0.0000	0.7180	0.0560
Baseline-Random	0.0583	0.5264	0.1050	0.2021	0.0220	0.5027

5.5.2 Evaluation under Cross-Domain Generalization

Table 5.11 presents model performance on the generalization split, where training and test samples come from different domains. Similar to the k-fold setting, the best results are achieved by the per-class variants of the tree-based models. The Random Forest per-class model attains the highest overall performance, with an **F2 score of 0.3667** and the best F1 score (**0.3643**), indicating strong generalization ability across strategy domains. The per-class Gradient Boosting model also performs well, achieving a slightly higher recall (**0.4040**) and F2 score (**0.3632**), but lower precision than Random Forest. Notably, generalization performance for both per-class models surpasses their k-fold counterparts, suggesting that domain-level generalization in narrative strategies may be more feasible than expected. This is a reversal of the trend seen in FEPD, where generalization typically resulted in reduced performance.

The multi-class models, while more conservative, perform worse overall. The Random Forest multi-class model achieves the highest precision (**0.4668**) and lowest Hamming Loss (**0.0660**), but very low recall (**0.0579**), leading to weak F2 performance (**0.0700**). The Gradient Boosting multi-class model also underperforms in recall and F2, reinforcing the advantages of per-class formulations for this task.

Table 5.11: Evaluation results for NSD task using the cross-domain generalization split.

Model Name	Precision	Recall	F1	F2	SubsetA.	H.Loss
GBC-multilabel	0.2962	0.1045	0.1541	0.1199	0.6249	0.0747
GBC-perclass	0.2595	**0.4040**	0.3157	0.3632	0.4950	0.1153
RandomForest-multilabel	**0.4668**	0.0579	0.1018	0.0700	**0.6583**	**0.0660**
RandomForest-perclass	0.3618	0.3686	**0.3643**	**0.3667**	0.5803	0.0841
VLM:Description	0.0143	0.0085	0.0106	0.0092	0.4967	0.1037
VLM:Definition	0.0122	0.0085	0.0100	0.0090	0.4599	0.1104
VLM:PerClass	0.0819	0.1949	0.1153	0.1527	0.4080	0.1968
Baseline-All	0.0658	1.0000	0.1234	0.2604	0.0000	0.9342
Baseline-None	0.0000	0.0000	0.0000	0.0000	0.6538	0.0658
Baseline-Random	0.0640	0.4958	0.1134	0.2110	0.0100	0.5100

As with the k-fold results, visual language model baselines continue to perform poorly overall. However, they appear largely unaffected by the domain shift: notably, the VLM:PerClass configuration improves its recall to **0.1949**, the highest recall observed among any VLM variant across both tasks. Despite this improvement, precision remains low (**0.0819**), and the model's F2 score (**0.1527**) remains below that of the random baseline. Interestingly, while the random baseline achieves a higher F2 score, the VLM:PerClass model now surpasses it in F1, indicating slightly more balanced performance. As expected, the heuristic baselines perform similarly to other settings, with the always-positive baseline again showing perfect recall but very low precision, and the always-negative baseline achieving relatively high subset accuracy due to conservative predictions.

5.5.3 Evaluation of Feature Sets for Narrative Strategies

To address RQ2, we evaluate how different feature subsets impact performance on the NSD task. Table 5.12 presents results under the k-fold setting for both per-class Gradient Boosting and per-class Random Forest across all feature configurations. The handcrafted feature set stands out as particularly strong, achieving the highest F1 score (**0.2730**) and F2 score (**0.3361**) with the Gradient Boosting Model. Its strength stems from a balanced combination of relatively high recall and precision. The multimodal-only subset with Gradient Boosting yields the highest recall (**0.4095**), but its extremely low precision (**0.0872**) leads to a much lower F2 score—mirroring patterns seen in the FEPD experiments.

The visual-only feature set also performs reasonably well (**F2** = **0.2888**), though less so than in FEPD, suggesting a reduced role for purely visual information in capturing narrative strategies. Audio-based subsets once again yield the weakest performance, followed by audio+multimodal and correlation-based sets. These results highlight that the effectiveness of a feature subset is less about its size and more about the relevance and quality of the features it includes.

Table 5.12: Feature experiment results for NSD on k-fold split.

Model	Featureset	Precision	Recall	F1	F2	SubsetA.	H.Loss
GBC	all	0.2343	0.3211	0.2709	0.2989	0.5742	0.0919
GBC	audio	0.1186	0.2041	0.1500	0.1783	0.4926	0.1230
GBC	visual	0.1786	0.3415	0.2345	0.2888	0.4961	0.1185
GBC	multimodal	0.0872	**0.4095**	0.1434	0.2346	0.2140	0.2620
GBC	audio&multi	0.1394	0.2639	0.1823	0.2237	0.5035	0.1265
GBC	correlations	0.1686	0.2721	0.2081	0.2422	0.5269	0.1105
GBC	handcrafted	0.2079	0.3973	**0.2730**	**0.3361**	0.5265	0.1125
RandomF.	all	0.2428	0.2190	0.2302	0.2234	**0.6445**	**0.0778**
RandomF.	audio	0.1413	0.1605	0.1501	0.1562	0.5807	0.0969
RandomF.	visual	0.2301	0.3238	0.2690	0.2994	0.5781	0.0935
RandomF.	multimodal	0.0848	0.3837	0.1390	0.2251	0.3173	0.2528
RandomF.	audio&multi	0.1818	0.2095	0.1945	0.2032	0.6059	0.0923
RandomF.	correlations	0.2019	0.1932	0.1973	0.1948	0.6306	0.0835
RandomF.	handcrafted	**0.2435**	0.2912	0.2651	0.2801	0.6007	0.0859

In the generalization setting (Table 5.13), overall performance improves for NSD, and the multimodal-only feature set continues to lead in recall (**0.6935**) and F2 (**0.4063**). However, the full feature sets offers the best balance between recall and precision, yielding the highest F1 scores. Interestingly, performance trends shift compared to the k-fold setup. The visual-only subset drops significantly in F1 and F2 scores, suggesting weaker generalization across domains for purely visual cues. The handcrafted feature set, which was highly effective in k-fold, performs much worse here and is surpassed by audio, audio+multimodal, and correlation-based feature sets in F1.

Table 5.13: Feature experiment results for NSD on cross-domain setup.

Model	**Featureset**	**Precision**	**Recall**	**F1**	**F2**	**SubsetA.**	**H. Loss**
GBC	all	0.2595	0.4040	0.3157	0.3632	0.4950	0.1153
GBC	audio	0.1645	0.2698	0.2041	0.2388	0.4532	0.1386
GBC	visual	0.1280	0.2599	0.1715	0.2154	0.3428	0.1650
GBC	multimodal	0.1542	**0.6935**	0.2516	**0.4063**	0.1182	0.2751
GBC	audio&multi	0.1985	0.3997	0.2652	0.3323	0.4153	0.1456
GBC	correlations	0.2289	0.3545	0.2781	0.3194	0.4353	0.1215
GBC	handcrafted	0.1437	0.3432	0.2025	0.2685	0.3094	0.1774
RandomF.	all	**0.3618**	0.3686	**0.3643**	0.3667	**0.5803**	**0.0841**
RandomF.	audio	0.2458	0.2669	0.2559	0.2624	0.5262	0.1021
RandomF.	visual	0.1162	0.1540	0.1323	0.1445	0.4086	0.1326
RandomF.	multimodal	0.1612	0.4336	0.2349	0.3239	0.2687	0.1855
RandomF.	audio&multi	0.2695	0.3460	0.3029	0.3274	0.4983	0.1047
RandomF.	correlations	0.2846	0.2500	0.2657	0.2560	0.5323	0.0908
RandomF.	handcrafted	0.1879	0.2020	0.1945	0.1989	0.4766	0.1097

In summary, NSD appears to rely more heavily on the interplay of multiple modalities than FEPD. While strong performance can be achieved using compact feature subsets—particularly those involving multimodal cues—the effectiveness of individual modalities like visual or audio remains inconsistent and highly task-dependent. Notably, visual features generalize worse than audio or multimodal ones, suggesting that different sources may diverge more in how they visually depict strategies, while audio and multimodal cues might remain more consistent across domains. These results reinforce the idea that in narrative strategy detection, the quality and relevance of features matter more than their quantity.

5.6 Model Performance Across Tasks

This section compares model performance across both prediction tasks—Film Editing Pattern Detection (FEPD) and Narrative Strategy Detection (NSD)—and evaluates the influence of model type, evaluation setup, and classification strategy. Specifically, we contrast tree-based classifiers (Random Forest and Gradient Boosting) with a zero-shot Visual Language Model (VLM), and assess differences between multi-class (joint) and per-class (independent) classification schemes. Each configuration is tested under both k-fold cross-validation and cross-domain generalization. All statistical analyses are conducted using non-parametric tests, as normality assumptions were not met. Results are reported at a significance threshold of $p < 0.05$.

5.6.1 Classifiers vs. Visual Language Models

To test the hypothesis that classifiers outperform vision-language models (VLMs), we conducted Mann–Whitney U-tests across both detection tasks and evaluation setups. Classifiers achieved significantly better results in all core metrics: micro-precision ($p = 0.001, U = 180$), micro-recall ($p = 0.0020, U = 163$), micro-F1 ($p = 0.0003, U = 174$), and micro-F2 ($p = 0.0010, U = 167$). While no significant differences were observed for subset accuracy or Hamming loss, these metrics are less informative in the context of imbalanced multi-label classification, where even trivial predictions can yield deceptively high scores. Overall, these findings suggest that the lightweight, zero-shot VLM evaluated here is not competitive with structured tree-based models when applied to localizing temporal patterns in news videos. The distribution of F2 scores across all tasks for both model types is shown in Figure 5.12.

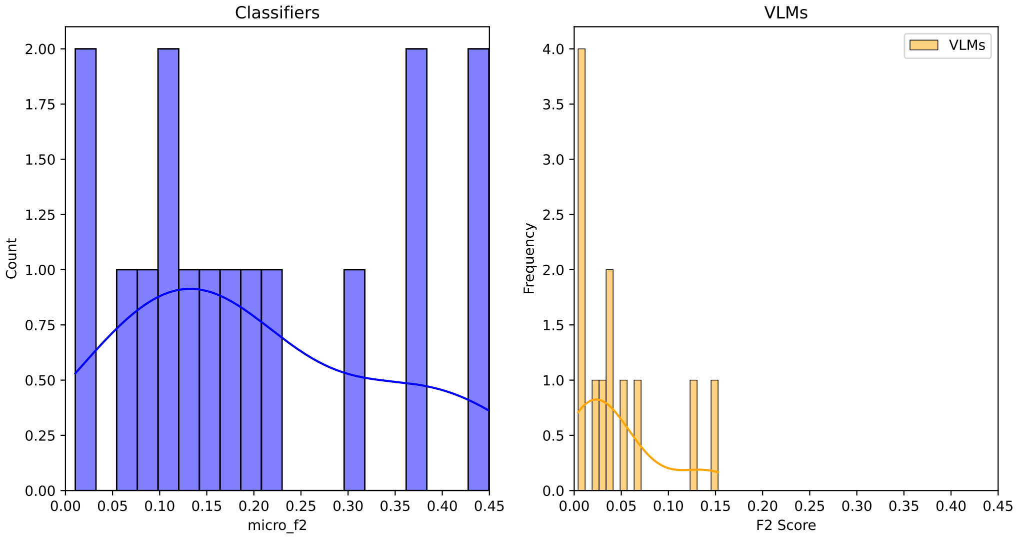


Figure 5.12: F2 score distribution across classifiers and VLMs. Classifiers consistently outperform VLMs.

5.6.2 Gradient Boosting vs. Random Forest

To compare the performance of the two classifier types, we conducted Wilcoxon signed-rank tests across all experiments. This test was appropriate as Gradient Boosting and Random Forest were always evaluated as matched pairs within each task. When including all feature configurations, Random Forests achieved significantly higher performance in:

- Micro-precision ($p = 0.0015, W = 409$)
- Subset accuracy ($p < 0.0001, W = 167.5$)
- Hamming loss (lower is better; $p = 0.0000, W = 74$)

In contrast, Gradient Boosting significantly outperformed Random Forests in:

- Micro-recall ($p = 0.0020, W = 402$)

- Micro-F2 ($p = 0.0493, W = 557$)

No significant difference was observed in micro-F1 scores ($p = 0.974$). These findings reflect a consistent trade-off between the two models: Random Forests make more conservative predictions with fewer false positives (higher precision), whereas Gradient Boosting is more effective at identifying true positives (higher recall), making it preferable when recall-focused metrics like F2 are prioritized.

Task- and Setup-Specific Comparisons. When analyzing only the narrative strategy detection task, the same pattern held: Random Forests significantly outperformed Gradient Boosting on micro-precision, subset accuracy, and Hamming loss, while Gradient Boosting performed significantly better on micro-recall and F2. In contrast, for the film editing pattern detection task, differences in precision- and recall-based metrics were not statistically significant; however, Random Forests still yielded significantly better subset accuracy and Hamming loss. This pattern also emerged when considering only the generalization evaluation setup.

When feature-based variants were excluded, overall trends persisted but with fewer significant differences—likely due to reduced sample size. Random Forests remained significantly better in Micro-precision ($p = 0.0234$), Subset accuracy ($p = 0.0078$) and Hamming loss ($p = 0.0078$). No significant differences were observed in micro-recall, micro-F1, or micro-F2.

While Random Forests achieved the highest F-scores in three out of four experiments that included all available features, this trend did not consistently generalize across all settings. Notably, in experiments involving feature selection, Gradient Boosting performed on par with—or in some cases better than—Random Forests. This suggests that Gradient Boosting may benefit more from reduced or optimized feature spaces.

5.6.3 Multi-Label vs. Per-Class Models

We compared the performance of models that predicted all labels at once and per-class models (independent binary classifiers per label) using Wilcoxon signed-rank tests. Per-class models significantly outperformed multi-label models in recall-focused metrics:

- Micro-recall ($p < 0.001$)
- Micro-F1 ($p < 0.001$)
- Micro-F2 ($p < 0.001$)

In contrast, multi-label models showed significantly better performance in:

- Subset accuracy ($p < 0.001$)
- Hamming loss ($p < 0.001$)

Differences in micro-precision were not significant when visual language models (VLMs) were included. However, when VLMs were excluded, multi-label models achieved significantly higher precision ($p = 0.0234$).

These results suggest a systematic trade-off: multi-label models tend to be more conservative, reducing false positives and yielding higher exact-match accuracy, while per-class models are more effective at capturing true positives—especially in recall-sensitive settings. Given the prioritization of recall in our target use case (e.g., via the F2 score), the per-class formulation is generally better suited for automatic preselection in a semi-automated analysis pipeline.

5.7 Discussion

The preceding sections presented a detailed exploration of both the dataset and the performance of various models on the tasks of detecting film editing patterns and narrative strategies in news video. This section reflects on those findings, drawing connections between the exploratory insights, feature behaviors, and model outcomes. Rather than summarizing results, the aim here is to interpret patterns, highlight implications, and consider the broader significance of the findings. We also identify points of friction—such as inconsistencies between expected and observed feature relevance—and reflect on the limitations of the current approach. Finally, we outline promising directions for future work that could build on and extend the insights developed here.

5.7.1 Pattern Exploration

The exploratory analysis revealed several noteworthy trends across both film editing patterns and narrative strategies.

Film Editing Patterns:

- **Alternating-shot** appeared across all outlets and exhibited robust correlations with shot similarity features, especially `prev2` and `next2`, which aligns with its definition. It was the longest pattern on average and frequently co-occurred with fragmentation-splitscreen and emotionalization.
- **Shot-reverse-shot** was also widely distributed and generally correlated more strongly with `prev2` and `next2` than with `prev1` and `next1`, particularly in terms of action similarity. Moderate correlations with emotional features and the multimodal interview feature support its presence in emotionally charged exchanges, especially in interview contexts.
- **Intensify** was found in all sources and showed negative correlations with shot similarity features, most notably `prev2` and `next2`. Interestingly, although Tagesschau generally uses the fewest patterns per story, this is the pattern it uses the most. Its average shot duration is significantly shorter than the overall mean, and it does not noticeably co-occur with other FEPs or strategies—suggesting it is a standalone pattern.
- **Cut-away** occurred less frequently but was evenly distributed. Roughly half of its instances co-occurred with alternating-shot, which aligns with the concept of visually cutting away and returning. It showed some correlation with the `EVENT` feature and associations with `prev2` and `next2`, which could indicate its partial use to cut away to events.
- **Cut-in** was similarly infrequent and characterized by the shortest average shot durations. Annotation deviations suggest slight inconsistencies in adhering to its formal definition. It showed no meaningful correlation with any features and only weak co-occurrence with

layperson individualization and emotionalization—possibly due to underrepresentation or noise.

Narrative Strategies:

- **Fragmentation-Splitscreen** stood out due to its exclusive use in Axel Springer outlets (Welt and BildTV). Unlike typical FEPs, it correlated more strongly with `next1` and `prev1` than with `next2` and `prev2`, possibly reflecting a distinct temporal structure.
- **Fragmentation** was more common in BildTV and CompactTV. It showed meaningful correlations with named entity tags, suggesting it is used in longer speech segments involving multiple entities or locations.
- **Emotionalization** was broadly distributed but did not correlate with either DeepFace-derived emotion features or the LLM-evaluative tag. This discrepancy raises questions about the validity of either the feature representations or the annotations.
- **Individualization of elite** appeared in all outlets except CompactTV and was the shortest narrative strategy on average. It was somewhat correlated with `eper`, as expected.
- **Individualization of reporter** was rare but primarily found in Welt. It was associated with long shots and sometimes overlapped with fragmentation-splitscreen, possibly reflecting outlet-specific production norms. Its strongest correlation was with active speaker, which is plausible given that the reporter is speaking and being shown.
- **Individualization of layperson** was infrequent and associated with shorter shots. It was almost always paired with emotionalization, supporting its conceptual role in making a situation more relatable and emotional to viewers by showing someone afflicted.

In general, public broadcasters made less use of FEPs per news story. Axel Springer channels used the most narrative strategies per story, largely driven by their frequent use of fragmentation-splitscreen. Overall, differences emerged not only between public and private broadcasters but also among individual channels—indicating that editorial style is shaped as much by outlet identity as by sector affiliation.

5.7.2 Feature Analysis

Besides the correlation analysis and feature set experiments, we initially intended to compute feature importance scores (e.g., via impurity decrease or permutation-based methods) to identify which features were most informative for pattern detection. However, given the relatively low overall classification performance—particularly with F1 scores consistently below 0.4—we concluded that such analyses would be of limited interpretive value and potentially misleading due to the lack of model reliability. Additionally, most model configurations included highly correlated features. In such cases, importance metrics can become unreliable, as importance is often arbitrarily distributed across redundant features. For these reasons, we opted against including feature importance experiments in this exploratory study. Instead, we focused on analyzing correlations between features and target patterns directly and interpreting the results of the feature set experiments, offering clearer and more interpretable insights.

Several overarching trends emerged from this approach. Within the feature space, highly similar variables—such as those from the shot similarity group (e.g., `prev1`, `prev2`, `next1`, `next2`) and various POS/NER tag counts—exhibited strong internal correlations. This internal consistency supports the construct validity of the extracted features. It also suggests that in subsequent analyses, not all of them need to be included, as many are functionally redundant.

For film editing pattern detection, visual features consistently outperformed other modalities across both k-fold and cross-domain evaluations. Shot similarity features, in particular, showed the strongest and most stable correlations with patterns like alternating-shot and intensify. This suggests that temporal visual rhythm and framing continuity are critical for recognizing film editing patterns. In contrast, audio-only features were consistently the weakest performers, and multimodal features—while occasionally useful—did not outperform purely visual ones. This highlights that for visually grounded constructs like FEPs, adding other modalities does not necessarily improve model performance and may even introduce noise or reduce generalizability.

For narrative strategy detection, the picture was more complex. The best-performing models often relied on either the strictly multimodal or handcrafted feature subsets, with handcrafted features sometimes yielding the most robust F1 scores overall. Multimodal features—despite being the smallest set (only seven variables)—occasionally achieved the highest recall, albeit often at the expense of precision. This suggests that feature quality can outweigh quantity in this domain. The relevance of multimodal signals reflects the more conceptually complex nature of narrative strategies, which often require information beyond visual shot elements.

Overall, the feature analysis underscores the importance of pattern-specific modeling. Shot-based features are well suited for visually defined editing patterns, while more abstract narrative strategies benefit from multimodal, deliberately selected inputs. In every experiment, feature set size was consistently outweighed by feature quality. Moving forward, the development of more discriminative, semantically grounded features—particularly for complex narrative strategies—could improve both detection performance and interpretability.

5.7.3 Model Performance and Generalization

Overall, the results confirm that temporal pattern detection remains a challenging task. While Hamming loss values were relatively low—reaching as little as 0.05 in some configurations—overall F1 and F2 scores remained moderate, underscoring that correctly predicting the appropriate patterns is substantially more difficult than simply avoiding false positives. Nevertheless, the strongest models achieved recall scores of nearly 70% in certain conditions, which is notable given the novelty of the task and the relatively small dataset.

Across both tasks, per-class models consistently outperformed multi-label models in recall-oriented metrics, including micro-recall, F1, and F2. In contrast, multi-label models yielded slightly higher precision and subset accuracy in some cases, reflecting a more conservative prediction style. This trade-off aligns with findings from the statistical tests: per-class models are better suited for recall-sensitive applications, while multi-label models tend to minimize false positives. Given the priority placed on recall in this exploratory context—particularly through the use of F2 as a key evaluation metric—the per-class approach appears better suited for semi-automated pattern detection pipelines.

Among the classifiers, both Random Forest and Gradient Boosting performed competitively, with no consistent winner across all experimental setups. Random Forests exhibited slightly more robust performance under domain shift and produced more conservative predictions, which translated into higher precision and lower Hamming loss. Gradient Boosting, on the other hand, achieved higher recall and F2 scores, particularly when feature selection was applied. This suggests that Gradient Boosting may benefit more from reduced or optimized feature spaces, while Random Forests are more resilient to noise and redundancy in the input data. Without feature selection, the per-class Random Forest configuration achieved the highest F-scores in three out of four experiments, highlighting its robustness in high-dimensional settings.

The lightweight Visual Language Model Qwen2.5-VL-3B-Instruct, evaluated as a zero-shot baseline, consistently underperformed relative to the tree-based models across all metrics. While its performance dropped less sharply under domain shift—likely due to the zero-shot setting—its absolute recall, precision, and F-scores remained considerably lower. Per-class prompting yielded some improvements over multi-label prompting in some cases, but no meaningful difference was observed between prompt types (i.e., definitions vs. descriptions). Overall, these results suggest that the VLM evaluated here is not yet suitable for fine-grained, temporal classification tasks in news videos. The generalizability of this finding to other VLMs is discussed in Section 5.7.4.

Finally, although generalization performance declined in FEPD when applied to unseen domains, some feature sets—particularly the visual and handcrafted subsets—exhibited slightly better cross-domain robustness, although their performance still worsened noticeably. For NSD, the inclusion of audio and multimodal features led to improved performance on the held-out channels compared to the k-fold setup. While this could partially reflect characteristics of the specific test channel, it suggests that audio and multimodal features may generalize better across outlets for NSD than visual features. In contrast, visual feature performance again declined in the cross-domain scenario, mirroring the FEPD results. This indicates that while visual information is generally important, its manifestation may vary more strongly across outlets. As such, thoughtful feature selection—particularly with regard to domain-invariant features—may play a critical role in improving model robustness across real-world settings.

5.7.4 Limitations

This work is the first to address the detection of both film editing patterns and narrative strategies in news videos, and as such, there was no existing benchmark to build upon. Several limitations follow from this.

Annotations and Definitions All annotations were produced by a single expert without inter-annotator agreement, introducing potential bias and limiting generalizability. Moreover, the definitions used to label patterns were not empirically validated prior to annotation. In practice, some annotations likely diverged from the formal definitions—e.g., patterns spanned fewer shots than specified or correlated with features in ways that contradicted theoretical expectations. Such inconsistencies may reflect subjective interpretation or definitional ambiguity. These issues risk systematic over- or underrepresentation of certain patterns in the training data, which can compromise both model performance and downstream analysis.

Dataset Contents The dataset is limited in size and diversity, covering just over seven hours of German-language news content from five outlets. Many patterns were rare, and some—though theoretically included—did not occur at all. This scarcity complicates model training, limits per-class and per-outlet comparisons, and highlights the overall challenge of detecting low-frequency stylistic devices. Furthermore, the exclusive use of German-language content limits cross-linguistic generalizability. Pre-trained components such as emotion recognition models or language encoders may perform worse on German than on English, which could have negatively impacted overall results. Additionally, journalistic and visual storytelling conventions vary by language and culture, further constraining the transferability of findings.

Model Configuration and Oversampeling The use of SMOTE to address class imbalance in the per-class setting introduces confounding effects. Performance improvements in these models may result from oversampling rather than from per-class modeling itself. Since SMOTE is not natively supported for multi-label classification, a direct comparison couldn't be drawn. It is possible that implementing a custom multi-label SMOTE variant could close this gap—especially since the multi-label models already showed stronger precision in many settings. Additionally, final hyperparameters differed significantly between models, so performance differences may reflect configuration choices as much as underlying model behavior.

Computational and Cost Constraints Hardware limitations restricted the scale and scope of model experiments, particularly for vision-language models. The Qwen2.5-VL-3B-Instruct model was selected for its accessibility, but significantly larger versions of Qwen—and stronger commercial alternatives—exist and often perform better. These were not tested due to computational and financial constraints. While such models might improve performance, they are also significantly more resource-intensive, which can be a minus not only in this research setting but also in real-world applications. In contrast, the tree-based classifiers used here are fast, lightweight, and compatible with interpretable feature sets—making them more viable in applied research or newsroom scenarios.

Cross-Domain Setup The cross-domain generalization experiment was conducted using only a single train-test split (training on four channels and testing on Welt). While this decision was theoretically motivated—Welt is the only outlet where all pattern types occur—it limits insight into how generalization might vary across other combinations. Future work could explore all possible 4-vs-1 configurations to assess robustness more thoroughly.

Feature Aggregation and Selection Despite the variety of features extracted, several potentially informative signals were not included—such as shot duration or inter-shot audio similarity. This is particularly relevant when comparing feature subsets: only the visual features contained relational cues, while others did not. Additionally, all features were aggregated at the span level, which may obscure important temporal or moment-level signals. For example, a speaker turn might begin in one shot and extend into another, but if emotional tone changes mid-way, both shots inherit the same aggregated features—potentially masking meaningful variation. Furthermore, all models treat spans as independent units and do not capture temporal dependencies across a single span, despite many patterns (e.g., *intensify*, *cut-in*) relying on such dynamics. No sequential, time-aware features (e.g., motion trajectories, emotional evolution, pacing curves) were included.

Metric Design and Evaluation Scope All evaluation metrics were based on multi-label classification logic and calculated at the shot level. While appropriate given the annotation structure, in real-world annotation workflows, approximate localization of patterns (e.g., finding a likely candidate segment rather than exact matches) may be more important than exact label alignment for all affected spans. Current metrics do not capture such flexibility. Moreover, no human-in-the-loop evaluation was conducted, so the practical utility of the models—as assistive tools in newsroom or research settings—remains untested.

5.7.5 Possibilities for Future Work

This thesis explored the automatic detection of film editing patterns and narrative strategies in German-language news videos. While the results offer a promising starting point, several opportunities remain for extending and refining the work. These fall into five main areas: pattern definition, dataset expansion, feature design, modeling, and evaluation.

Pattern Taxonomy Development and Expansion The pattern taxonomy introduced in Section 3.1.2 forms the foundation of this work. However, this foundation remains unstable: the definitions have never been empirically validated or cross-checked, and may therefore lack clarity or consistency. Especially in the context of automation, clearly defined and unified annotation criteria are essential for building a reliable dataset and benchmarking model performance. They are also crucial for enabling generalization across languages and media systems. Some patterns may benefit from subdivision, while others might require merging or redefinition based on annotation behavior and model confusion patterns.

Expanding and Diversifying the Dataset One of the clearest paths forward is to expand the annotated dataset—both in terms of size, diversity and quality. A larger dataset would enable more robust learning and facilitate the use of more complex, data-hungry models such as deep neural networks. Beyond simply increasing the volume of data, diversifying content across languages and cultural contexts would help assess the generalizability of the models and the underlying taxonomy. In particular, including English-language content would additionally allow integration of stronger pre-trained components and support clearer comparisons with related work in media studies and NLP. Additionally, inter-annotator studies should be conducted to assess the reliability of the current taxonomy. Empirical validation of the pattern definitions—through small-scale experiments with trained annotators—would help refine the criteria and reduce inconsistencies in future annotations.

Extending the Feature Set and Modeling of Context At present, span-level predictions mainly treat inputs as isolated units, without modeling most surrounding context or temporal dependencies. Introducing temporal modeling—e.g., using temporal convolutions or sequence-based architectures—could improve detection by capturing transitions and dependencies across spans. Given the inherently temporal nature of many patterns, such modeling could yield substantial improvements.

The findings also indicate that visual features are most informative for detecting editing patterns, while narrative strategies require a more balanced multimodal approach. Future work could focus on two main areas: first, improving the coverage and granularity of visual features; and second, extending the multimodal feature space with carefully engineered features explicitly tailored to the narrative strategy definitions.

Alternative Models and Learning Setups While tree-based classifiers were effective in certain subtasks and partly interpretable, future work could explore more complex models, particularly those suited for multimodal and sequential data. This includes transformer-based architectures with cross-modal attention, as well as models pre-trained on video-language tasks but fine-tuned on the specific data. More powerful vision-language models may also be evaluated, ideally with tuning or adaptation to the annotation scheme. Moreover, implementing a multi-label variant of SMOTE—or alternative sampling techniques that respect label co-occurrence—could improve model performance in the multi-label setting, where recall remains comparatively weak.

Human-in-the-Loop Evaluation Given the applied nature of the task, an important direction for future work is integrating these models into assistive annotation tools and evaluating their performance in human-in-the-loop scenarios. The goal would not be to fully automate annotation, but to support it—e.g., by flagging likely candidate spans or suggesting pattern labels. Such tools could be tested in collaboration with journalists or communication researchers to evaluate usability, interpretability, and time-saving potential. Evaluation metrics could then be adapted to better reflect practical needs, such as partial matches, ranked predictions, or calibrated confidence scores—helping to bridge the gap between academic modeling and real-world annotation workflows.

6 Conclusion

This thesis explored the occurrence and detection of temporal patterns—specifically Film Editing Patterns and Narrative Strategies —in news videos. Recognizing the growing importance of video-based media and the persuasive power of stylistic framing, this work adopted an interdisciplinary perspective, integrating insights from media studies, computer vision, and machine learning. By addressing three central research questions, the study provides empirical insights and technical benchmarks for the automated analysis of these patterns. This conclusion synthesizes the key findings and reflects on their broader implications.

6.1 RQ1: Pattern Exploration

How are film editing patterns and narrative strategies distributed across news outlets, and how do they co-occur or relate to each other?

The exploratory analysis revealed clear structural differences between public and private news broadcasters—both in overall pattern density and in the types of patterns used. Generally, public broadcasters used the least temporal patterns per news story. While some patterns, like *shot-reverse-shot* or *emotionalization*, were found across all outlets, others were highly outlet-specific, suggesting editorial or ideological preferences. Notably, there were not just differences between public and private broadcasters but also stark contrasts between private channels from different owners. The most striking example was the exclusive use of *fragmentation-splitscreen* by the Springer-owned outlets (Welt and BildTV); this pattern never appeared in public broadcasters nor in CompactTV. In contrast, *fragmentation* (without splitscreen) occurred almost exclusively in private outlets. *Cut-away* was predominantly used by CompactTV, while the public channels hardly employed it. *Intensify* was the most common film editing pattern found in public broadcasters. These findings align with Bateman and Tseng [13], which suggests that not only the frequency but also the types of strategies used vary systematically between news sources.

The co-occurrence analysis further supported these differences: specific film editing patterns and narrative strategies often appeared together—such as *cut-away* with *alternating-shot*, or *emotionalization* with *individualization of layperson.* Some of these co-occurrences are likely due to structural similarities or shared outlet preferences, but others—like the frequent pairing of layperson individualization with emotionalization—suggest more meaningful editorial patterns. This particular link reinforces that laypeople are often used as emotionally resonant figures in news storytelling.

6.2 RQ2: Feature Analysis

Which types of features—visual, audio, linguistic, or multimodal—are most informative for detecting film editing patterns and narrative strategies, and how does feature effectiveness vary by task?

This thesis employed a wide range of interpretable features derived from visual, auditory, and textual data. Across experiments, visual features were consistently the most informative for detecting film editing patterns—an expected result, given that editing techniques are inherently visual in nature. In contrast, narrative strategies benefited most from a balanced multimodal approach. Handcrafted and multimodal feature sets yielded the highest performance, highlighting that narrative strategies are multimodal in nature. Multimodal feature sets especially outperformed visual features in the narrative strategy cross-domain task. These findings underscore the importance of combining modalities—particularly for generalization—since patterns may differ more in their visual form across outlets than in their underlying structure or content. Additionally, smaller, well-designed sets often outperformed larger ones, revealing a consistent trend toward quality over quantity. For narrative strategies in particular, targeted feature selection improved recall compared to using the full feature set—especially if guided by the annotation definitions themselves.

Taken together, these results highlight the value of interpretable features in video analysis. Regardless of model performance, interpretable features are essential for diagnostics, transparency, and use in media-critical or assistive contexts.

6.3 RQ3: Model Performance

How well do machine learning models detect these patterns, and what are their limitations across settings or domains?

This thesis systematically evaluated both per-pattern binary classifiers and multi-label approaches using Random Forest and Gradient Boosting Classifiers. Across nearly all metrics, especially recall and F1—the per-pattern models outperformed their multi-label counterparts. However, this advantage must be interpreted with caution: the per-class models also benefited from SMOTE-based oversampling, which was not available in the multi-label setting. As for model comparison, there was no clear winner between Random Forest and Gradient Boosting Classifiers across all conditions. Broadly, Random Forest tended to yield higher precision, while Gradient Boosting Classifiers offered better recall. That said, in three out of four full-feature experiments, the per-class Random Forest achieved the best F-scores—highlighting its robustness in high-dimensional or imbalanced settings.

Models were also evaluated under two setups: standard k-fold cross-validation and a more challenging cross-domain generalization setting. As expected, performance generally dropped under domain shift. However, the relative ranking of models remained largely stable. Notably, this generalization trend held for film editing patterns but did not apply as strongly to narrative strategies—where some per-class models actually performed better in the domain-shift condition. This suggests that while generalization remains difficult, certain patterns and models may transfer more robustly than others.

The vision-language model (Qwen2.5-VL-3B-Instruct), evaluated as a zero-shot baseline, performed poorly across all settings. Despite strong results on general video-language tasks, it failed to detect the structured, temporally grounded patterns targeted in this work—regardless of prompt type. This points to a key limitation of current general-purpose VLMs: While they are impressive in their broad applicability, they may struggle with precise, multimodal classification tasks unless explicitly adapted. That said, only one VLM was tested here, so further experiments with larger or fine-tuned models are needed before drawing strong conclusions.

6.4 Final Remarks

This thesis demonstrates that detecting film editing patterns and narrative strategies in news videos is not only feasible but also insightful when using interpretable, feature-based models. Although performance varied depending on the type of pattern and the features used, the experiments show that even relatively simple, transparent models can capture meaningful editorial choices. This opens up possibilities for developing tools that help researchers, educators, and media professionals better understand and analyze visual news content.

Furthermore, by pursuing this interpretable route of detection, we also gained deeper insights into the patterns themselves. We examined the temporal pattern from many angles, their occurrence by news outlet, their relations with each other, and their relations with the interpretable features. Through this, we developed a more nuanced understanding of how these patterns function in journalistic practice. The results reveal clear stylistic differences between outlets and offer insight into how meaning is shaped not only by what is said but by how it is visually presented and structured.

At the same time, the work also underscores key challenges. Model performance was often constrained by limited labeled data and the subjective or ambiguous nature of some pattern definitions. Looking ahead, improving annotations, expanding datasets, and incorporating human-in-the-loop approaches will be essential for building more reliable and generalizable systems. Overall, this thesis provides a solid foundation for further work in this area. It shows that even with modest resources, it is possible to shed light on the often temporal structures that shape how news is depicted, and in doing so, it contributes to a more transparent and critically engaged media landscape.

Bibliography

[1] Jake K. Aggarwal and Quin Cai. "Human Motion Analysis: A Review". In: *Computer Vision and Image Understanding* 73.3 (1999), pp. 428–440. DOI: 10.1006/CVIU.1998.0744.

[2] Pravesh Agrawal et al. "Pixtral 12B". In: *arXiv preprint* abs/2410.07073 (2024). DOI: 10.48550/ARXIV.2410.07073.

[3] Esma Aımeur, Sabrine Amri, and Gilles Brassard. "Fake news, disinformation and misinformation in social media: a review". In: *Soc. Netw. Anal. Min.* 13.1 (2023), p. 30. DOI: 10.1007/S13278-023-01028-5.

[4] Jean-Baptiste Alayrac et al. "Flamingo: a Visual Language Model for Few-Shot Learning". In: *Annual Conference on Neural Information Processing Systems, NeurIPS 2022, New Orleans, USA, November 28 - December 9, 2022.* 2022. DOI: 10.1109/SP46215.2023.10179434.

[5] Anthropic. *Claude.* https://www.anthropic.com/claude. [Accessed 23-03-2025].

[6] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. "ViViT: A Video Vision Transformer". In: *International Conference on Computer Vision, ICCV 2021, Montreal, Canada, October 10-17, 2021.* IEEE, 2021, pp. 6816–6826. DOI: 10.1109/ICCV48922.2021.00676.

[7] Arun Babu et al. "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale". In: *Annual Conference of the International Speech Communication Association, ISCA 2022, Incheon, Korea, September 18-22, 2022.* ISCA, 2022, pp. 2278–2282. DOI: 10.21437/INTERSPEECH.2022-143.

[8] Adam Badawy, Emilio Ferrara, and Kristina Lerman. "Analyzing the Digital Traces of Political Manipulation: The 2016 Russian Interference Twitter Campaign". In: *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018, Barcelona, Spain, August 28-31, 2018.* IEEE Computer Society, 2018, pp. 258–265. DOI: 10.1109/ASONAM.2018.8508646.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.* 2015. URL: http://arxiv.org/abs/1409.0473.

[10] Shuai Bai et al. "Qwen2.5-VL Technical Report". In: *arXiv preprint* abs/2502.13923 (2025). DOI: 10.48550/ARXIV.2502.13923.

[11] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. "WhisperX: Time-Accurate Speech Transcription of Long-Form Audio". In: *Annual Conference of the International Speech Communication Association, ISCA 2023, Dublin, Ireland, August 20-24, 2023.* ISCA, 2023, pp. 4489–4493. DOI: 10.21437/INTERSPEECH.2023-78.

[12] Alexandra Balahur, Ralf Steinberger, Mijail A. Kabadjov, Vanni Zavarella, Erik Van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. "Sentiment Analysis in the News". In: *International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta.* European Language Resources Association, 2010. URL: http://www.lrec-conf.org/proceedings/lrec2010/summaries/909.html.

[13] John A Bateman and Chiao-I Tseng. "Multimodal discourse analysis as a method for revealing narrative strategies in news videos". In: *Multimodal Communication* 12.3 (2023), pp. 261–285. DOI: 10.1515/mc-2023-0029.

[14] Steven S. Beauchemin and John L. Barron. "The Computation of Optical Flow". In: *Association for Computing Machinery Computing Surveys* 27.3 (1995), pp. 433–467. DOI: 10.1145/212094.212141.

[15] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. "Is Space-Time Attention All You Need for Video Understanding?" In: *International Conference on Machine Learning, ICML 2021, Virtual Event, 18-24 July, 2021*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 813–824. URL: http://proceedings.mlr.press/v139/bertasius21a.html.

[16] David Bordwell. "Intensified Continuity Visual Style in Contemporary American Film". In: *Film Quarterly* 55 (2002), pp. 16–28. DOI: 10.1525/FQ.2002.55.3.16.

[17] Kurt Braddock and James Price Dillard. "Meta-analytic evidence for the persuasive effect of narratives on beliefs, attitudes, intentions, and behaviors". In: *Communication monographs* 83.4 (2016), pp. 446–467. DOI: 10.1080/03637751.2015.1128555.

[18] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[19] Dipanita Chakraborty, Werapon Chiracharit, and Kosin Chamnongthai. "Video shot boundary detection using principal component analysis (pca) and deep learning". In: *International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2021, Chiang Mai, Thailand, May 19–22, 2021*. IEEE. 2021, pp. 272–275. DOI: 10.1109/ECTI-CON51831.2021.9454775.

[20] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357. DOI: 10.1613/JAIR.953.

[21] Gullal S Cheema, Judi Arafat, Chiao-I Tseng, John A Bateman, Ralph Ewerth, and Eric Müller-Budack. "Identification of Speaker Roles and Situation Types in News Videos". In: *International Conference on Multimedia Retrieval, ICMR 2024, Phuket, Thailand, June 10–14, 2024*. 2024, pp. 506–514. DOI: 10.1145/3652583.3658101.

[22] Gullal Singh Cheema. *videomae-base-finetuned-kinetics-movieshots-multitask*. https://huggingface.co/gullalc/videomae-base-finetuned-kinetics-movieshots-multitask. [Accessed 26-03-2025].

[23] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. "BEATs: Audio Pre-Training with Acoustic Tokenizers". In: *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 5178–5193. URL: https://proceedings.mlr.press/v202/chen23ag.html.

[24] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *International Conference on Knowledge Discovery and Data Mining, ACM SIGKDD 2016, San Francisco, USA, August 13 - 17, 2016*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[25] *COMPACTTV*. https://www.youtube.com/channel/UCgvFsn6bRKqND1cW3HpzDrA. [Accessed 29-03-2025].

[26] Matthew Cooper, Ting Liu, and Eleanor Gilbert Rieffel. "Video Segmentation via Temporal Pattern Classification". In: *IEEE Transactions on Multimedia* 9.3 (2007), pp. 610–618. DOI: 10.1109/TMM.2006.888015.

[27] Corinna Cortes and Vladimir Vapnik. “Support-Vector Networks”. In: *Machine Learning* 20.3 (1995), pp. 273–297. DOI: 10.1007/BF00994018.

[28] Peter Dahlgren. *Media and political engagement: citizens, communication, and democracy.* English. United Kingdom: Cambridge University Press, 2009. ISBN: 978-0-521-52789-7.

[29] Michael F. Dahlstrom. “Using narratives and storytelling to communicate science with nonexpert audiences”. In: *Proceedings of the National Academy of Sciences* 111.supplement_4 (2014), pp. 13614–13620. DOI: 10.1073/pnas.1320645111. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1320645111.

[30] Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuoling Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. *NVLM: Open Frontier-Class Multimodal LLMs.* 2024. DOI: 10.48550/ARXIV.2409.11402.

[31] Navneet Dalal and Bill Triggs. “Histograms of Oriented Gradients for Human Detection”. In: *Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005, San Diego, CA, USA 20-26 June, 2005.* IEEE Computer Society, 2005, pp. 886–893. DOI: 10.1109/CVPR.2005.177.

[32] Viorela Dan, Britt Paris, Joan Donovan, Michael Hameleers, Jon Roozenbeek, Sander van der Linden, and Christian von Sikorski. “Visual mis-and disinformation, social media, and democracy”. In: *Journalism & Mass Communication Quarterly* 98.3 (2021), pp. 641–664. DOI: 10.1177/10776990211035395.

[33] Michael Han Daniel Han and Unsloth team. *Unsloth.* 2023. URL: http://github.com/unslothai/unsloth.

[34] Matt Deitke et al. *Molmo and PixMo: Open Weights and Open Data for State-of-the-Art Multimodal Models.* 2024. DOI: 10.48550/ARXIV.2409.17146.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019.* Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/V1/N19-1423.

[36] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Trevor Darrell, and Kate Saenko. “Long-term recurrent convolutional networks for visual recognition and description”. In: *Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015.* IEEE Computer Society, 2015, pp. 2625–2634. DOI: 10.1109/CVPR.2015.7298878.

[37] Khaoula Elagouni, Christophe Garcia, and Pascale Sébillot. “A comprehensive neural-based approach for text recognition in videos using natural language processing”. In: *International Conference on Multimedia Retrieval, ICMR 2011, Trento, Italy, April 18 - 20, 2011.* ACM, 2011, p. 23. DOI: 10.1145/1991996.1992019.

[38] Joseph G. Ellis, Brendan Jou, and Shih-Fu Chang. “Why We Watch the News: A Dataset for Exploring Sentiment in Broadcast Video News”. In: *International Conference on Multimodal Interaction, ICMI 2014, Istanbul, Turkey, November 12-16, 2014.* ACM, 2014, pp. 104–111. DOI: 10.1145/2663204.2663237.

[39] Jeffrey L. Elman. “Finding Structure in Time”. In: *Cognitive Science* 14.2 (1990), pp. 179–211. DOI: 10.1207/S15516709COG1402_1.

[40] Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric J. Topol, Jeff Dean, and Richard Socher. “Deep learning-enabled medical computer vision”. In: *npj Digital Medicine* 4 (2021). DOI: 10.1038/S41746-020-00376-2.

[41] Alef Iury Siqueira Ferreira. *WAV2VEC2-large-xlsr-53-gender-recognition-librispeech*. https://huggingface.co/alefiury/wav2vec2-large-xlsr-53-gender-recognition-librispeech. [Accessed 26-03-2025].

[42] James H. Fetzer. "Disinformation: The Use of False Information". In: *Minds and Machines* 14.2 (2004), pp. 231–240. DOI: 10.1023/B:MIND.0000021683.28604.5B.

[43] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451.

[44] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. "Vision meets robotics: The KITTI dataset". In: *International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237. DOI: 10.1177/0278364913491297.

[45] Junaid Ahmed Ghauri, Sherzod Hakimov, and Ralph Ewerth. "Supervised Video Summarization Via Multiple Feature Sets with Parallel Attention". In: *IEEE International Conference on Multimedia and Expo, ICME 2021, Shenzhen, China, July 5-9, 2021*. IEEE, 2021, 1–6s. DOI: 10.1109/ICME51207.2021.9428318.

[46] DWDL.de GmbH. *Experiment vorbei: Axel Springer beendet Bild TV zum Jahresende - DWDL.de — dwdl.de*. https://www.dwdl.de/nachrichten/95709/experiment_vorbei_axel_springer_beendet_bild_tv_zum_jahresende/. [Accessed 29-03-2025].

[47] Yihong Gong and Xin Liu. "Video Summarization Using Singular Value Decomposition". In: *Conference on Computer Vision and Pattern Recognition CVPR 2000, Hilton Head, SC, USA, 13-15 June, 2000*. IEEE Computer Society, 2000, pp. 2174–2180. DOI: 10.1109/CVPR.2000.854772.

[48] Lucas Graves and Federica Cherubini. "The rise of fact-checking sites in Europe". In: *Digital News Project Report* (2016). DOI: 10.60625/risj-tdn4-p140.

[49] Sorin Mihai Grigorescu, Bogdan Trasnea, Tiberiu T. Cocias, and Gigel Macesanu. "A survey of deep learning techniques for autonomous driving". In: *Journal Field Robotics* 37.3 (2020), pp. 362–386. DOI: 10.1002/ROB.21918. URL: https://doi.org/10.1002/rob.21918.

[50] Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans-Joachim Böhme. "Training a Broad-Coverage German Sentiment Classification Model for Dialog Systems". In: *Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*. European Language Resources Association, 2020, pp. 1627–1632. URL: https://aclanthology.org/2020.lrec-1.202/.

[51] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. "Creating Summaries from User Videos". In: *Computer Vision, ECCV 2014, Zurich, Switzerland, September 6-12, 2014*. Vol. 8695. Lecture Notes in Computer Science. Springer, 2014, pp. 505–520. DOI: 10.1007/978-3-319-10584-0_33.

[52] Anne Hamby, Hongmin Kim, and Francesca Spezzano. "Sensational stories: The role of narrative characteristics in distinguishing real and fake news and predicting their spread". In: *Journal of Business Research* 170 (2024), p. 114289. DOI: https://doi.org/10.1016/j.jbusres.2023.114289.

[53] C. Happer and Greg Philo. "The Role of the Media in the Construction of Public Belief and Social Change". In: *Journal of Social and Political Psychology* 1 (2013), pp. 321–336. DOI: 10.5964/JSPP.V1I1.96.

[54] Alexander G. Hauptmann and Michael J. Witbrock. "Story Segmentation and Detection of Commercials in Broadcast News Video". In: *Forum on Research and Technology Advances in Digital Libraries, IEEE-ADL 1998, Santa Barbara, USA, April 22-24, 1998*. IEEE Computer Society, 1998, pp. 168–179. DOI: 10.1109/ADL.1998.670392.

[55] Tin Kam Ho. “Random decision forests”. In: *Conference on Document Analysis and Recognition, ICDAR 1995, August 14 - 15, 1995, Montreal, Canada, 1995.* IEEE Computer Society, 1995, pp. 278–282. DOI: 10.1109/ICDAR.1995.598994.

[56] Sepp Hochreiter. “Untersuchungen zu dynamischen neuronalen Netzen”. In: *Diploma, Technische Universität München* 91.1 (1991), p. 31.

[57] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computing* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/NECO.1997.9.8.1735.

[58] Berthold K. P. Horn and Brian G. Schunck. “Determining Optical Flow”. In: *Artificial Intelligence* 17.1-3 (1981), pp. 185–203. DOI: 10.1016/0004-3702(81)90024-2.

[59] Winston H. Hsu, Shih-Fu Chang, Chih-Wei Huang, Lyndon S. Kennedy, Ching-Yung Lin, and Giridharan Iyengar. “Discovery and fusion of salient multimodal features toward news story segmentation”. In: *Storage and Retrieval Methods and Applications for Multimedia 2004, San Jose, CA, USA, January 20, 2004.* Vol. 5307. SPIE Proceedings. SPIE, 2004, pp. 244–258. DOI: 10.1117/12.533037.

[60] Mahesh G. Huddar, Sanjeev S. Sannakki, and Vijay S. Rajpurohit. “A Survey of Computational Approaches and Challenges in Multimodal Sentiment Analysis”. In: *International Journal of Computer Sciences and Engineering* 7.1 (2019), pp. 876–883. DOI: 10.26438/ijcse/v7i1.876883.

[61] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. “Large-Scale Video Classification with Convolutional Neural Networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014.* IEEE Computer Society, 2014, pp. 1725–1732. DOI: 10.1109/CVPR.2014.223.

[62] Zied Kechaou, Ali Wali, Mohamed Ben Ammar, Hichem Karray, and Adel M. Alimi. “A novel system for video news’ sentiment analysis”. In: *Journal of Systems and Information Technology* 15.1 (2013), pp. 24–44. DOI: 10.1108/13287261311322576.

[63] Maheshkumar H Kolekar and S Sengupta. “Semantic indexing of news video sequences: a multimodal hierarchical approach based on hidden markov model”. In: *IEEE Tencon, 2005, Melbourn, Australia, 21-24 November, 2005.* IEEE. 2005, pp. 1–6. DOI: 10.1109/TENCON.2005.301204.

[64] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Annual Conference on Neural Information Processing Systems, NeurIPS 2012, December 3-6, 2012, Lake Tahoe, Nevada, United States, 2012.* 2012, pp. 1106–1114. DOI: 10.1145/3065386.

[65] Jiyoung Lee, Michael Hameleers, and Soo Yun Shin. “The emotional effects of multimodal disinformation: How multimodality, issue relevance, and anxiety affect misperceptions about the flu vaccine”. In: *New Media & Society* 26.12 (2024), pp. 6838–6860. DOI: 10.1177/14614448231153959.

[66] Stephan Lewandowsky, Ullrich KH Ecker, Colleen M Seifert, Norbert Schwarz, and John Cook. “Misinformation and its arXiv preprintection: Continued influence and successful debiasing”. In: *Psychological science in the public interest* 13.3 (2012), pp. 106–131. DOI: 10.1177/1529100612451018.

[67] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models”. In: *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA.* Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 19730–19742. URL: https://proceedings.mlr.press/v202/li23q.html.

[68] Yang Liu. "Initial Study on Automatic Identification of Speaker Role in Broadcast News Speech". In: *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*. The Association for Computational Linguistics, 2006. URL: `https://aclanthology.org/N06-2021/`.

[69] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. "Video Swin Transformer". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 3192–3201. DOI: `10.1109/CVPR52688.2022.00320`.

[70] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110. DOI: `10.1023/B:VISI.0000029664.99615.94`.

[71] Bruce D. Lucas and Takeo Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision". In: *International Joint Conference on Artificial Intelligence, IJCAI 1981, Vancouver, BC, Canada, August 24-28, 1981*. William Kaufmann, 1981, pp. 674–679.

[72] Joseph Magliano and Jeffrey M. Zacks. "The Impact of Continuity Editing in Narrative Film on Event Segmentation". In: *Cognitive Science* 35.8 (2011), pp. 1489–1517. DOI: `10.1111/J.1551-6709.2011.01202.X`.

[73] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. "Unsupervised Video Summarization with Adversarial LSTM Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 2982–2991. DOI: `10.1109/CVPR.2017.318`.

[74] Maxwell E McCombs and Donald L Shaw. "The agenda-setting function of mass media". In: *Public opinion quarterly* 36.2 (1972), pp. 176–187. DOI: `10.1075/asj.1.2.02mcc`.

[75] Meta. *Llama 3.2: Revolutionizing edge ai and vision with open, customizable models.* `https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/`. [Accessed 23-03-2025].

[76] Joanne M. Miller and J. Krosnick. "News Media Impact on the Ingredients of Presidential Evaluations: Politically Knowledgeable Citizens Are Guided by a Trusted Source". In: *American Journal of Political Science* 44 (2000), pp. 301–315. DOI: `10.2307/2669312`.

[77] Google Deep Mind. *Gemini*. `https://deepmind.google/technologies/gemini/`.

[78] D. Mobbs, N. Weiskopf, H. Lau, E. Featherstone, R. Dolan, and C. Frith. "The Kuleshov Effect: the influence of contextual framing on emotional attributions." In: *Social cognitive and affective neuroscience* 1 2 (2006), pp. 95–106. DOI: `10.1093/SCAN/NSL014`.

[79] Markus Mühling, Manja Meister, Nikolaus Korfhage, Jörg Wehling, Angelika Hörth, Ralph Ewerth, and Bernd Freisleben. "Content-based video retrieval in historical collections of the German Broadcasting Archive". In: *International Journal on Digital Libraries* 20.2 (2019), pp. 167–183. DOI: `10.1007/S00799-018-0236-Z`.

[80] Hans-Hellmut Nagel. "Displacement vectors derived from second-order intensity variations in image sequences". In: *Computer Vision, Graphics, and Image Processing* 21.1 (1983), pp. 85–117. DOI: `10.1016/S0734-189X(83)80030-9`.

[81] Nic Newman, Richard Fletcher, Craig T Robertson, Amy Ross Arguedas, and Rasmus Kleis Nielsen. "Reuters Institute digital news report 2024". In: *Reuters Institute for the study of Journalism* (2024).

[82] Nic Newman, Richard Fletcher, Anne Schulz, Simge Andi, Craig T Robertson, and Rasmus Kleis Nielsen. "Reuters Institute digital news report 2021". In: *Reuters Institute for the study of Journalism* (2021).

[83] Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. "Beyond short snippets: Deep networks for video classification". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015.* IEEE Computer Society, 2015, pp. 4694–4702. DOI: `10.1109/CVPR.2015.7299101`.

[84] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. *Expanding Language-Image Pretrained Models for General Video Recognition.* 2022. DOI: `10.1007/978-3-031-19772-7\_1`.

[85] "Object Recognition from Local Scale-Invariant Features". In: *International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999.* IEEE Computer Society, 1999, pp. 1150–1157. DOI: `10.1109/ICCV.1999.790410`.

[86] OpenAI. `https://platform.openai.com/docs/models/gpt-4o`. [Accessed 20-03-2025].

[87] OpenGVLab. *INTERNVL*. `https://github.com/OpenGVLab/InternVL`. [Accessed 20-03-2025].

[88] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. DOI: `10.5555/1953048.2078195`.

[89] Moisés Henrique Ramos Pereira, Flávio Luis Cardeal Pádua, Adriano César Machado Pereira, Fabrício Benevenuto, and Daniel Hasan Dalip. "Fusing Audio, Textual, and Visual Features for Sentiment Analysis of News Videos". In: *International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.* AAAI Press, 2016, pp. 659–662. URL: `http://www.aaai.org/ocs/index.php/ICWSM/ICWSM16/paper/view/13144`.

[90] Dinh Quoc Phung, Chitra Dorai, and Svetha Venkatesh. "Narrative Structure Analysis with Education and Training Videos for E-Learning". In: *Conference on Pattern Recognition, ICPR 2002, Quebec, Canada, August 11-15, 2002.* IEEE Computer Society, 2002, p. 835. DOI: `10.1109/ICPR.2002.1048432`. URL: `https://doi.org/10.1109/ICPR.2002.1048432`.

[91] Ronald Poppe. "A survey on vision-based human action recognition". In: *Image and Vision Computing* 28.6 (2010), pp. 976–990. DOI: `10.1016/J.IMAVIS.2009.11.014`.

[92] Ethan Porter and Thomas J. Wood. "The global effectiveness of fact-checking: Evidence from simultaneous experiments in Argentina, Nigeria, South Africa, and the United Kingdom". In: *Proceedings of the National Academy of Sciences* 118.37 (2021), e2104235118. DOI: `10.1073/pnas.2104235118`.

[93] Gert-Jan Poulisse, Marie-Francine Moens, Tomas Dekens, and Koen Deschacht. "News story segmentation in multiple modalities". In: *Multimedia Tools and Applications* 48.1 (2010), pp. 3–22. DOI: `10.1007/S11042-009-0358-9`.

[94] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". In: *Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020.* Association for Computational Linguistics, 2020, pp. 101–108. DOI: `10.18653/V1/2020.ACL-DEMOS.14`.

[95] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *International Conference on Machine Learning, ICML 2021, Virtual Event, 18-24 July, 2021.* Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763. URL: `http://proceedings.mlr.press/v139/radford21a.html`.

[96] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. "Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking". In: *Conference on Empirical Methods in Natural Language Processing, 2017, Copenhagen, Denmark*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2931–2937. DOI: 10.18653/v1/D17-1317.

[97] Mickael Rouvier, Sebastien Delecraz, Benoıt Favre, Meriem Bendris, and Frédéric Béchet. "Multimodal embedding fusion for robust speaker role recognition in video broadcast". In: *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015, Scottsdale, AZ, USA, December 13-17, 2015*. IEEE, 2015, pp. 383–389. DOI: 10.1109/ASRU.2015.7404820.

[98] Patricia Salvador and Miguel Cobos. "Narrative as a Key Element for Learning Through Videos". In: *IEEE Seventh Ecuador Technical Chapters Meeting, (ECTM) 2023, Ambato,10th-13th October, 2023*. IEEE. 2023, pp. 1–5. DOI: 10.1109/ETCM58927.2023.10309024.

[99] Richard J Schaefer and Tony J Martinez III. "Trends in network news editing strategies from 1969 through 2005". In: *Journal of Broadcasting & Electronic Media* 53.3 (2009), pp. 347–364. DOI: 10.1080/08838150903102600.

[100] S. Schwan, Bärbel Garsoffky, and F. Hesse. "Do film cuts facilitate the perceptual and cognitive organization of activitiy sequences?" In: *Memory Cognition* 28 (2000), pp. 214–223. DOI: 10.3758/BF03213801.

[101] Sefik Serengil and Alper Ozpinar. "A Benchmark of Facial Recognition Pipelines and Co-Usability Performances of Modules". In: *Journal of Information Technologies* 17.2 (2024), pp. 95–107. DOI: 10.17671/gazibtd.1399077.

[102] Sefik Ilkin Serengil and Alper Ozpinar. "LightFace: A Hybrid Deep Face Recognition Framework". In: *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE. 2020, pp. 23–27. DOI: 10.1109/ASYU50717.2020.9259802.

[103] Sefik Ilkin Serengil and Alper Ozpinar. "HyperExtended LightFace: A Facial Attribute Analysis Framework". In: *International Conference on Engineering and Emerging Technologies, ICEET 2021, Istanbul, Turkey, October 27–28, 2021*. IEEE. 2021, pp. 1–4. DOI: 10.1109/ICEET53442.2021.9659697.

[104] Karen Simonyan and Andrew Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *Conference on Neural Information Processing Systems, NeurIPS 2014, Montreal,Canada, December 8-13, 2014*. 2014, pp. 568–576. URL: https://proceedings.neurips.cc/paper/2014/hash/00ec53c4682d36f5c4359f4ae7bd7ba1-Abstract.html.

[105] Mohammad Soleymani, David García, Brendan Jou, Björn W. Schuller, Shih-Fu Chang, and Maja Pantic. "A survey of multimodal sentiment analysis". In: *Image and Vision Computing* 65 (2017), pp. 3–14. DOI: 10.1016/J.IMAVIS.2017.08.003.

[106] Tomás Soucek and Jakub Lokoc. "TransNet V2: An Effective Deep Network Architecture for Fast Shot Transition Detection". In: *ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*. ACM, 2024, pp. 11218–11221. DOI: 10.1145/3664647.3685517.

[107] Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. "RoFormer: Enhanced transformer with Rotary Position Embedding". In: *Neurocomputing* 568 (2024), p. 127063. DOI: 10.1016/J.NEUCOM.2023.127063.

[108] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. "VideoBERT: A Joint Model for Video and Language Representation Learning". In: *IEEE International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 7463–7472. DOI: 10.1109/ICCV.2019.00756.

[109] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. "VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training". In: *Conference on Neural Information Processing Systems, NeurIPS 2022, New Orleans, USA, November 28 - December 9, 2022.* 2022. URL: `http://papers.nips.cc/paper%5C_files/paper/2022/hash/416f9cb3276121c42eebb86352a4354a-Abstract-Conference.html`.

[110] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "Learning Spatiotemporal Features with 3D Convolutional Networks". In: *IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015.* IEEE Computer Society, 2015, pp. 4489–4497. DOI: `10.1109/ICCV.2015.510`.

[111] Chiao-I Tseng, John A Bateman, Leandra Thiele, Ralph Ewerth, Eric Müller-Budack, Gullal Cheema, Manuel Burghardt, and Bernhard Liebl. "The search for filmic narrative strategies in audiovisual news reporting: a progress report". In: *Society for Cognitive Studies of the Moving Image Conference (SCSMI).* Budapest, Hungary, June 2024.

[112] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Conference on Neural Information Processing Systems,NeurIPS 2017, Long Beach, USA, December 4-9, 2017.* 2017, pp. 5998–6008. URL: `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

[113] Alessandro Vinciarelli and Sarah Favre. "Broadcast news story segmentation using social network analysis and hidden markov models". In: *International Conference on Multimedia, ACM-MM 2007, Augsburg, Germany, September 24-29, 2007.* ACM, 2007, pp. 261–264. DOI: `10.1145/1291233.1291287`.

[114] S. Waisbord. "Truth is What Happens to News". In: *Journalism Studies* 19 (2018), pp. 1866–1878. DOI: `10.1080/1461670X.2018.1492881`.

[115] Peter Wittenburg, Hennie Brugman, Albert Russel, Alexander Klassmann, and Han Sloetjes. "ELAN: a Professional Framework for Multimodality Research". In: *International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006.* European Language Resources Association (ELRA), 2006, pp. 1556–1559. URL: `http://www.lrec-conf.org/proceedings/lrec2006/summaries/153.html`.

[116] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. "ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders". In: *arXiv preprint* abs/2301.00808 (2023). DOI: `10.48550/arXiv.2301.00808`.

[117] Hui-Yin Wu, Francesca Palù, Roberto Ranon, and Marc Christie. "Thinking Like a Director: Film Editing Patterns for Virtual Cinematographic Storytelling". In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 14.4 (2018), 81:1–81:22. DOI: `10.1145/3241057`.

[118] Fu Xiaoling and Gao Hua. "Gray-based news video text extraction approach". In: *International Conference on Computer Sciences and Convergence Information Technology ICCIT 2010, Seoul, South Korea, 30 November - 2 December, 2010.* 2010, pp. 208–211. DOI: `10.1109/ICCIT.2010.5711058`.

[119] An Yang et al. "Qwen2.5 Technical Report". In: *arXiv preprint* abs/2412.15115 (2024). DOI: `10.48550/ARXIV.2412.15115`.

[120] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. "Defending Against Neural Fake News". In: *Conference on Neural Information Processing Systems, NeurIPS 2019, Vancouver, BC, Canada, December 8-14, 2019.* 2019, pp. 9051–9062. URL: `https://proceedings.neurips.cc/paper/2019/hash/3e9f0fc9b2f89e043bc6233994dfcf76-Abstract.html`.

[121] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. *Sigmoid Loss for Language Image Pre-Training.* 2023. DOI: 10.1109/ICCV51070.2023.01100.

[122] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Places: A 10 Million Image Database for Scene Recognition". In: *IEEETransactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1452–1464. DOI: 10.1109/TPAMI.2017.2723009.

A Appendix Chapter 1

A.1 Film Editing Pattern Definitions

- Actor Continuity: Focuses on the same actor(s) over several shots.

  ```
  Actor Continuity {
      length: >= 2
      actor-relation: same (all)
  }
  ```

- Alternating Shots A (Actors): Alternates between two distinct (groups of) actors across multiple shots, with each appearing at least twice.

  ```
  alternating-shots-actors {
      length: >= 4
      actor-relation: same (s1, s3), same (s2, s4)
      place-relation: same (s1, s3), same (s2, s4)
      visual-similarity: low (s1, s2)
      place-relation: different (s1,s2)
  }
  ```

- Alternating Shots B (Objects): Alternates between two distinct (sets of) objects across multiple shots, with each appearing at least twice.

  ```
  alternating-shots-objects {
      length: >= 4
      object-relation: same (s1, s3), same (s2, s4)
      place-relation: same (s1, s3), same (s2, s4)
      visual-similarity: low (s1, s2)
  }
  ```

- Alternating Shots C (Objects + Actor): Cuts from one or multiple object(s) to one or multiple actor(s), then returns to the same object(s) before returning again to the same actor(s).

  ```
  alternating-shots-objects-actors {
      length: >= 4
      object-relation: same (s1, s3)
      actor-relation: same (s2, s4)
      place-relation: same (s1, s3), same (s2, s4)
      visual-similarity: low (s1, s2)
  }
  ```

- Alternating Shots D (Actors + Objects): Cuts from one or multiple actor(s) to one or multiple object(s), then goes back to the same actor(s) and object(s).

  ```
  alternating-shots-actors-objects {
      length: >= 4
      actor-relation: same (s1, s3)
      object-relation: same (s2, s4)
      place-relation: same (s1, s3), same (s2, s4)
  ```

```
    visual-similarity: low (s1, s2)
}
```

- Ambiance Enhancement: Maintains consistent sound or music for at least three consecutive shots.

```
Ambience Enhancement {
    length: >= 3
    music/sound-relation: same (all)
}
```

- Continuity of Talk/Dialogue: Preserves dialogue continuity across multiple shots, even if the speaker is not shown.

```
Continuity of Talk {
    length: >= 2
    speaker-relation: same (all)
}
```

- Cut-away Version 1 (To People): Momentarily shifts focus from the main subject to a group of people (often for context or reaction) and then returns to the original subject.

```
cut-away-v1 {
    length: == 3
    actor-relation: different (s1, s2)
    actor-relation: same (s1, s3)
    visual-similarity: low (s1, s2)
    visual-similarity: high (s1, s3)
    num-faces: >= 2 (s2)

}
```

- Cut-away Version 2 (To Objects): Temporarily shifts to one or multiple object(s) by inserting a different shot and then returns to the original subject.

```
cut-away-v2 {
    length: >= 3
    actor-count: 0 (s2)
    actor-relation: same (s1, s3)
    EITHER
        place-relation: same (s1, s2)
        size-relation: closer (s1, s2)
    OR
        place-relation: different (s1, s2)
}
```

- Cut-in: Briefly transitions from a wider shot to a closer view or detail of the same scene (e.g., part of an object), then moves on or returns.

```
cut-in {
    length: == 3
    actor-relation: same (s1, s3)
    size-relation: closer (s1, s2)
    object-relation: same (s1, s2)
}
```

- Double Cut-in: A variant of the cut-in involving two consecutive close-ups of the same object for emphasis.

```
double-cut-in {
    length: >= 4
    actor-relation: same (s1, s4)
    size-relation: closer (s2, s3)
    object-relation: same (all)
}
```

- Frameshare: Consecutive shots of different actors occupying the same on-screen region, typically to suggest agreement or alignment between them.

```
frameshare {
    length: = 2
    actor-count: 1 (s1, s2)
    actor-relation: different (s1, s2)
    region-relation: same (s1, s2)
}
```

- Intensify Version 1 (Same Face): Progressively zooms in on the same face over a sequence of shots.

```
intensify-v1 {
    length: >= 3
    actor-relation: same (all)
    size-relation: closer (all)
}
```

- Intensify Version 2 (Multiple Faces): Progressively zooms in on multiple faces over a sequence of shots.

```
intensify-v2 {
    length: >= 3
    face-count: >= 3 (all)
    size-relation: closer (all)
    average-face-size <= 0.3 (all)
    EITHER
        semantic-similarity: high (s1, s2, s3)
    OR
        Visual-similarity: high (s1, s2)
        actor-relation: same, at least 1 actor (s2, s3)

}
```

- Intensify Version 3 (Object): Progressively zooms in on an object over a sequence of shots.

```
intensify-v3 {
    length: >= 2
    object-relation: same (all)
    size-relation: closer (all)
}
```

- Opposition: Consecutive shots of different actors framed in different on-screen regions, often conveying disagreement or contrasting viewpoints.

```
opposition {
    length: = 2
    actor-count: 1 (s1, s2)
    actor-relation: different (s1, s2)
    region-relation: different (s1, s2)
}
```

- POV: Shows the point of view of two different actors by having a medium or closer shot size and alternating from one actor to the other and back.

  ```
  POV {
      length: >= 3
      shot-size: medium or closer (s1,s3)
      actor-relation: different (s1, s2)
      actor-relation: same (s1, s3)
  }
  ```

- Shot Reverse Shot: Repeatedly alternates between two characters (e.g., in conversation), commonly used to depict dialogue or direct interaction.

  ```
  shot-reverse-shot {
      length: >= 3
      actor-count: 1 (all)
      actor-relation: different (s1, s2)
      actor-relation: same (s1, s3)
      gaze: non-direct (all)
  }
  ```

- Spatial Continuity: Depicts the same location over several shots.

  ```
  Continuity of Talk {
      length: >= 2
      place-relation: same (all)
  }
  ```

- Thematic Enhancement: Emphasizes a theme by repeating a particular action or movement (e.g., walking, queueing) across multiple shots.

  ```
  Thematic Enhancement {
      length: >= 3
      movement-relation: same (all)
  }
  ```

A.2 Vision-Language Model Prompts

```
FILM_EDITING_PROMPT = """
### Context
You are analyzing film editing patterns. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
1. Determine if the **current shot** matches one or more of the following editing
patterns.
2. If no pattern applies, return an empty list.
3. Output the result as a JSON object with the key `"patterns"`.

### Editing Patterns
1. **alternating-shot**:
```

```
   - Alternates between two distinct actors, groups, or objects across multiple
   shots.
   - Each actor/group/object appears at least twice.

2. **cut-away**:
   - Temporarily shifts focus from the main subject (e.g., an actor) to a
   secondary element (e.g., a group or object).
   - Then returns to the original subject.

3. **cut-in**:
   - Transitions from a wider shot to a closer detail of the same scene or subject.
   - Then returns to the original framing or context.

4. **intensify**:
   - Gradually zooms in (or moves closer) on a face or object over a sequence of
   shots.

5. **shot-reverse-shot**:
   - Alternates between two characters, typically in dialogue.
   - Focuses on close or medium shots of each character as they speak or react.

### Output Format
Return **only** a JSON object with the key `"patterns"`. Example outputs:
{
  "patterns": ["alternating-shot", "cut-away"]
}
or
{
  "patterns": []
}"""



FILM_EDITING_DEFS_PROMPT = """
### Context
You are analyzing film editing patterns. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
1. Determine if the **current shot** matches one or more of the following editing
patterns.
2. If no pattern applies, return an empty list.
3. Output the result as a JSON object with the key `"patterns"`.

Below are the patterns with formal definitions. Each bullet describes constraints or
relations among shots (s1, s2, s3, ...). For example, “actor-relation: same (s1, s3)”
means the same actor (or set of actors) appears in shot 1 and shot 3.

### Editing Patterns and Their Definitions:
```

```
**alternating-shot**
- Length:  4
- Actor-or-object-relation: same (s1, s3), same (s2, s4)
- Place-relation: same (s1, s3), same (s2, s4)
- Visual-similarity: low (s1, s2)

**cut-away**
- Length:  3
- Actor-relation: same (s1, s3)
- Must satisfy one of the following conditions:
  1. Actor-count:  2 (s2)
     Actor-relation: different (s1, s2)
     Visual-similarity: low (s1, s2)
     Visual-similarity: high (s1, s3)
  2. Actor-count: 0 (s2)
     Place-relation: same (s1, s2)
     Size-relation: closer (s1, s2)
  3. Actor-count: 0 (s2)
     Place-relation: different (s1, s2)

**cut-in**
- Length: = 3
- Actor-relation: same (s1, s3)
- Size-relation: closer (s1, s2)
- Object-relation: same (s1, s2)

**intensify**
- Length:  2
- Actor-or-object-relation: same (all)
- Size-relation: closer (all)

**shot-reverse-shot**
- Length:  3
- Actor-count: 1 (all)
- Actor-relation: different (s1, s2)
- Actor-relation: same (s1, s3)
- Gaze: non-direct (all)

### Output Format
Return **only** a JSON object with the key `"patterns"`. Example outputs:
{
  "patterns": ["alternating-shot", "cut-away"]
}
or
{
  "patterns": []
}
"""


ALTERNATING_SHOT = """
```

```
### Context
You are analyzing film editing patterns. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of an alternating-shots pattern.
Output a boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: At least 4 shots.
- Alternates between two distinct actors or objects.
- Each actor or object appears at least twice in alternating positions.
- Visual similarity is low between alternating shots (e.g., shot 1 and shot 2).

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)

### Example Outputs
1
0
"""


CUT_AWAY = """
### Context
You are analyzing film editing patterns. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of a cut-away pattern. Output a
boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: Typically 3 shots, but may vary slightly.
- Shifts focus from the main actor or subject in shot 1 to a secondary element
(e.g., a group, object, or detail) in shot 2.
- Returns to the original actor or subject in shot 3.
- EITHER:
  - The secondary element in shot 2 is visually similar to shot 1 but closer in
  framing (e.g., a detail of the same scene).
 OR:
  - The secondary element in shot 2 is visually distinct, showing a different
```

```
    location, context, or focus.
- Visual similarity is low between shot 1 and shot 2, but high between shot 1 and
shot 3.
- Shot 2 often contains multiple faces, focuses on an object, or depicts a non-actor
element.

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)

### Example Outputs
1
0
"""



CUT_IN = """
### Context
You are analyzing film editing patterns. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of a cut-in pattern.
Output a boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: Typically 3 shots, but may vary slightly.
- Begins with a wider shot (shot 1), transitions to a closer view or detail of
the same scene or object in shot 2, and returns to the original framing in shot 3.
- The object or scene remains consistent across all shots.
- Shot 2 is visually closer than shot 1.

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)

### Example Outputs
1
0
"""




INTENSIFY = """
### Context
You are analyzing film editing patterns. Inputs include:
```

```
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of an intensify pattern.
Output a boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: At least 2 shots, but often extends to several shots.
- Focuses on the same actor or object across all shots.
- Gradually zooms in or moves closer to the actor or object over the sequence.
- Visual framing becomes progressively tighter.

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)

### Example Outputs
1
0
"""


SHOT_REVERSE_SHOT = """
### Context
You are analyzing film editing patterns. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of a shot-reverse-shot pattern.
Output a boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: At least 3 shots, but often extends to depict ongoing dialogue.
- Alternates between two characters, typically in conversation.
- Each shot focuses on one character, with low visual similarity between
alternating shots.
- Characters’ gazes are non-direct (e.g., looking off-screen toward the other
character).

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)
```

```
### Example Outputs
1
0
"""

STRATEGIES_PROMPT = """
### Context
You are analyzing media patterns in visual and verbal content. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
1. Determine if the **current shot** matches one or more of the following patterns.
2. If no pattern applies, return an empty list.
3. Output the result as a JSON object with the key `"patterns"`.

### Media Patterns
1. **fragmentation**:
   - Splits visual and/or verbal information to highlight specific story elements
   or multiple perspectives.
   - Alternates shots of a speaker with shots that do not show the speaker, creating
   breaks in visual continuity.

2. **fragmentation_splitscreen**:
   - Splits visual and/or verbal information to highlight specific story elements or
   multiple perspectives.
   - Uses a split-screen format to show simultaneous perspectives or actions.

3. **individualization_of_elite**:
   - Focuses on a single elite individual-someone with power or high public standing
   -rather than referencing a group or institution (e.g., “the government”).
   - Shows this elite person in a studio context where they are identified by name,
   followed by a shot of them in a different (non-studio) setting.
   This emphasizes their personal agency or responsibility.

4. **individualization_of_reporter**:
   - Centers on a reporter in a situation outside the news studio.
   - The reporter references themselves (e.g., “My opinion is...”), moving from a
   neutral information source to a more personal and possibly emotional figure.

5. **individualization_of_layperson**:
   - Focuses on someone from the general public-an everyday individual. The viewer
   sees and hears this person, giving them presence and voice within the narrative.
   - Typically shows the layperson in a non-studio context (e.g., at home or
   work), emphasizing their personal experience and distinguishing them from
   professional media figures.

6. **emotionalization**:
   - Heightens the emotional tone of a topic.
   - Displays the same emotion (e.g., sadness, shock, excitement) across multiple
   shots to underscore its significance in the narrative.

### Output Format
Return **only** a JSON object with the key `"patterns"`. Example outputs:
{
  "patterns": ["fragmentation", "individualization_of_layperson"]
}
or
{
  "patterns": []
}
"""

STRATEGIES_DEFS_PROMPT = """
### Context
You are analyzing media patterns in visual and verbal content. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
1. Determine if the **current shot** matches one or more of the following patterns.
2. If no pattern applies, return an empty list.
3. Output the result as a JSON object with the key `"patterns"`.

Below are the patterns with formal definitions. Each bullet describes constraints or
relations among shots (s1, s2, s3, ...). For example, “actor-relation: same (s1, s3)”
means the same actor (or set of actors) appears in shot 1 and shot 3.

### Media Patterns and Their Definitions:

**fragmentation**
- length: >= 4
- speaker-relation: same (all)
- shot1 + shot3: actor-relation = same, place-relation = same, talkspace = on-screen
- shot2 + shot4: object-relation = same, place-relation = same, talkspace = off-screen
- shot1 + shot2: visual-similarity = low

**fragmentation_splitscreen**
- length: = 1
- actor.role: != reporter
- talkspace: on-screen
- splitscreen: true

**individualization_of_elite**
- length: >= 2
- shot1: place = studio, talkspace = on-screen, spoken_text >= 1 named entity (person)
- shot2: place != studio, talkspace = off-screen, actor.role != {reporter, anchor}
- actor-relation: same (all)

**individualization_of_layperson**
- length: >= 2
- place: != studio
- last_shot: actor.role != {reporter, anchor}, talkspace = on-screen
- actor-relation: at least one actor same (all)

**individualization_of_reporter**
- length: >= 1
- last_speaker_turn: spoken_text >= 1 self-referent-pronoun
- last_shot: talkspace = on-screen
- place != studio
- actor.role = reporter

**emotionalization**
- length: >= 2
EITHER
   face-emotion: not neutral
   face-emotion-relation: same (all)
OR
   sentiment: highly positive/negative


### Output Format
Return **only** a JSON object with the key `"patterns"`. Example outputs:
{
  "patterns": ["fragmentation", "individualization_of_layperson"]
}
or
{
  "patterns": []
}
"""


FRAGMENTATION = """
### Context
You are analyzing media patterns in visual and verbal content. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of a fragmentation pattern.
Output a boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: At least 4 shots.
- The speaker remains the same across all shots.
- Shot 1 and shot 3 show the speaker in the same place.
- Shot 2 and shot 4 show the same object in the same place but differ from shot 1
in visual content.

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)

### Example Outputs
1
0
"""

FRAGMENTATION_SPLITSCREEN = """
### Context
You are analyzing media patterns in visual and verbal content. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of a fragmentation_splitscreen pattern.
Output a boolean value (1 or 0).

### Pattern Definition
A single shot where:
- The actor is not a reporter.
- The speaker is on-screen.
- The shot uses a split-screen format to show simultaneous perspectives or actions.

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)

### Example Outputs
1
0
"""

INDIVIDUALIZATION_OF_ELITE = """" "
### Context
You are analyzing media patterns in visual and verbal content. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.
```

```
### Task
Determine whether the **current shot** is part of an individualization_of_elite pattern.
Output a boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: At least 2 shots.
- Shot 1 shows an elite individual in a studio context, identified by name.
- Shot 2 shows the same elite individual in a different setting.
- Both shots focus on the same elite individual.

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)

### Example Outputs
1
0
"""

INDIVIDUALIZATION_OF_LAYPERSON = """
### Context
You are analyzing media patterns in visual and verbal content. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of an individualization_of_layperson
pattern. Output a boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: At least 2 shots.
- The place is not a studio.
- The last shot does not feature a reporter or anchor and has on-screen talkspace.
- At least one actor remains the same across all shots.

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)

### Example Outputs
1
0
"""
INDIVIDUALIZATION_OF_REPORTER = """
### Context
```

```
You are analyzing media patterns in visual and verbal content. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of an individualization_of_reporter
pattern. Output a boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: At least 1 shot.
- Has a reporter talking on-screen outside a studio setting.
- The reporter references themselves in the spoken text.

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)

### Example Outputs
1
0
"""

EMOTIONALIZATION = """
### Context
You are analyzing media patterns in visual and verbal content. Inputs include:
- The two previous shots (if available).
- The current shot (to classify).
- The next two shots (if available).
- Dialogue transcripts for each speaker-turn during the shots.

### Task
Determine whether the **current shot** is part of an emotionalization pattern.
Output a boolean value (1 or 0).

### Pattern Definition
A sequence of shots where:
- Length: At least 2 shots.
- EITHER:
    - The face emotion is not neutral across all shots.
    - The face emotion remains the same across all shots.
- OR:
    - The sentiment is highly positive or negative.

### Output Format
Return **only** a boolean value:
1 (match)
0 (no match)
```

```
### Example Outputs
1
0
"""
```

A.3 Shot Length Distributions

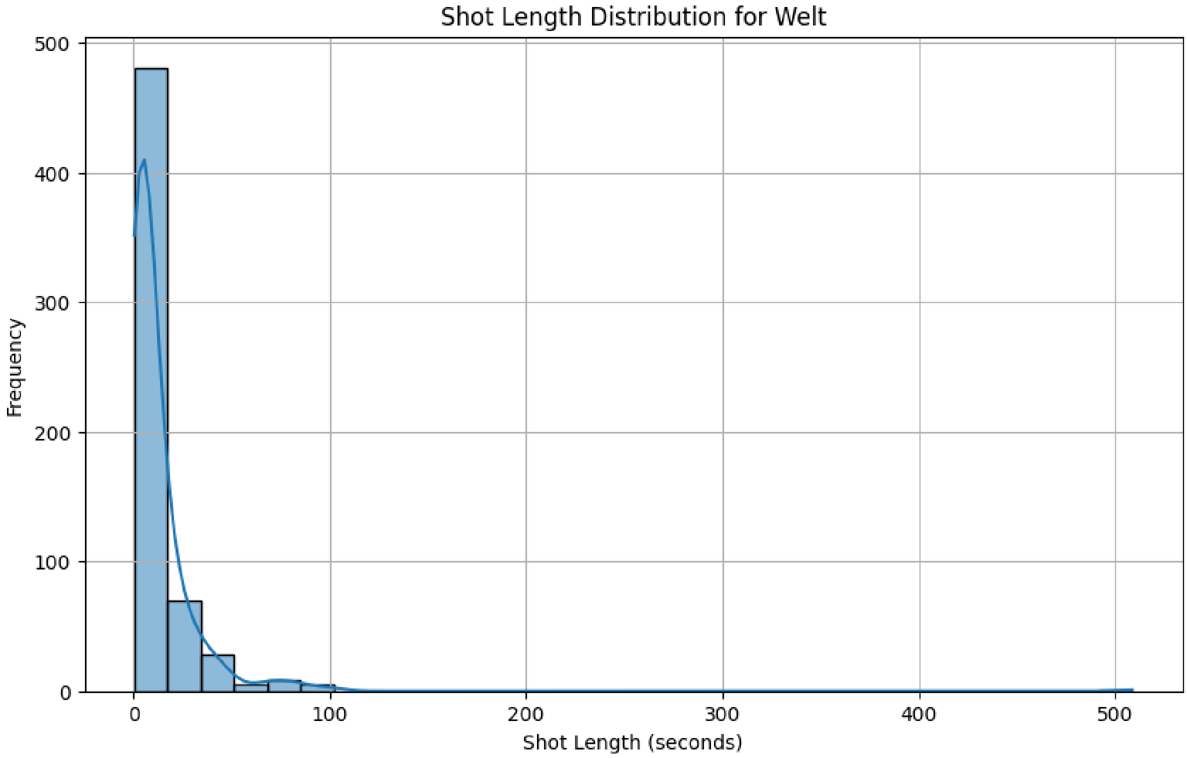


Figure A.1: Shot length distribution for Welt.

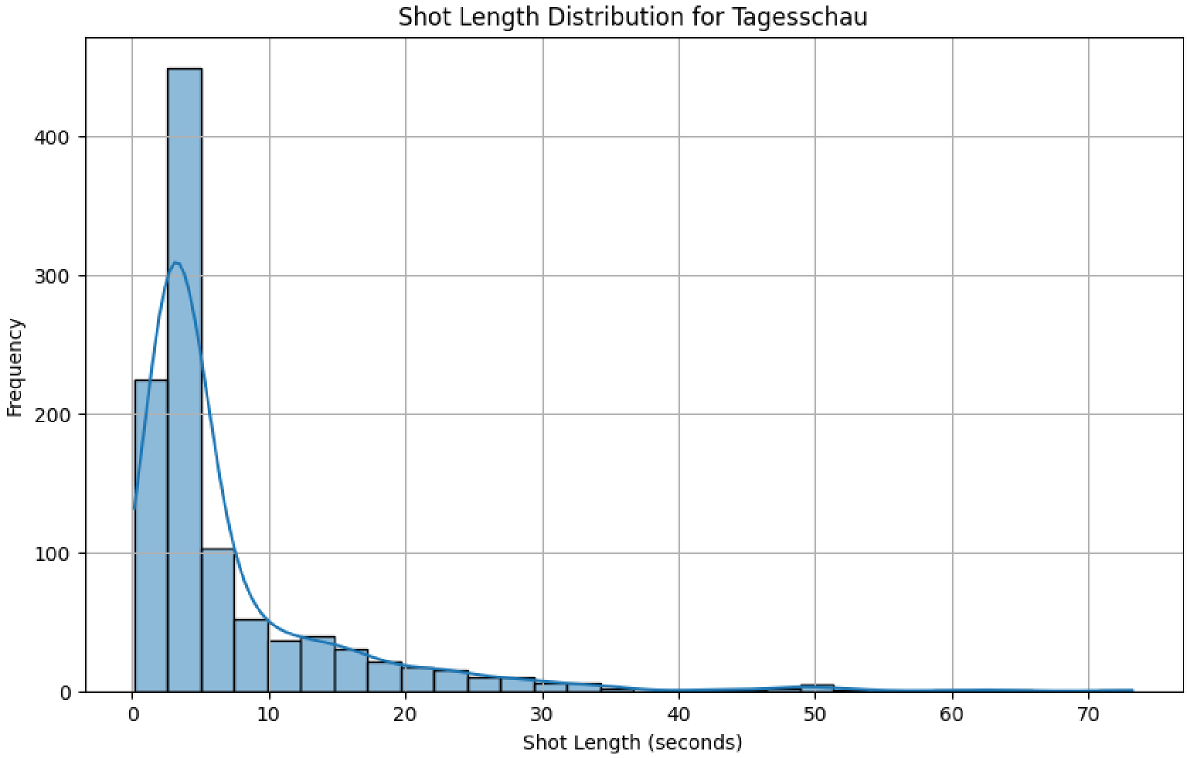


Figure A.2: Shot length distribution for Tagesschau.

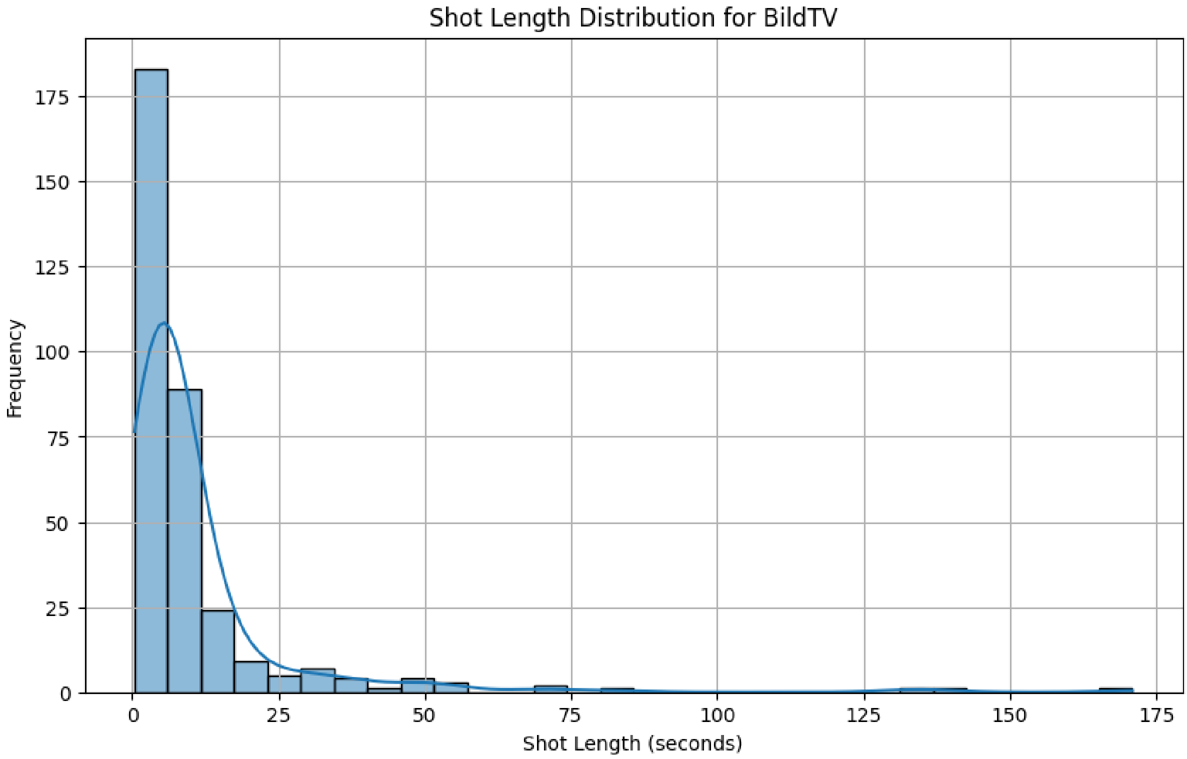


Figure A.3: Shot length distribution for BildTV.

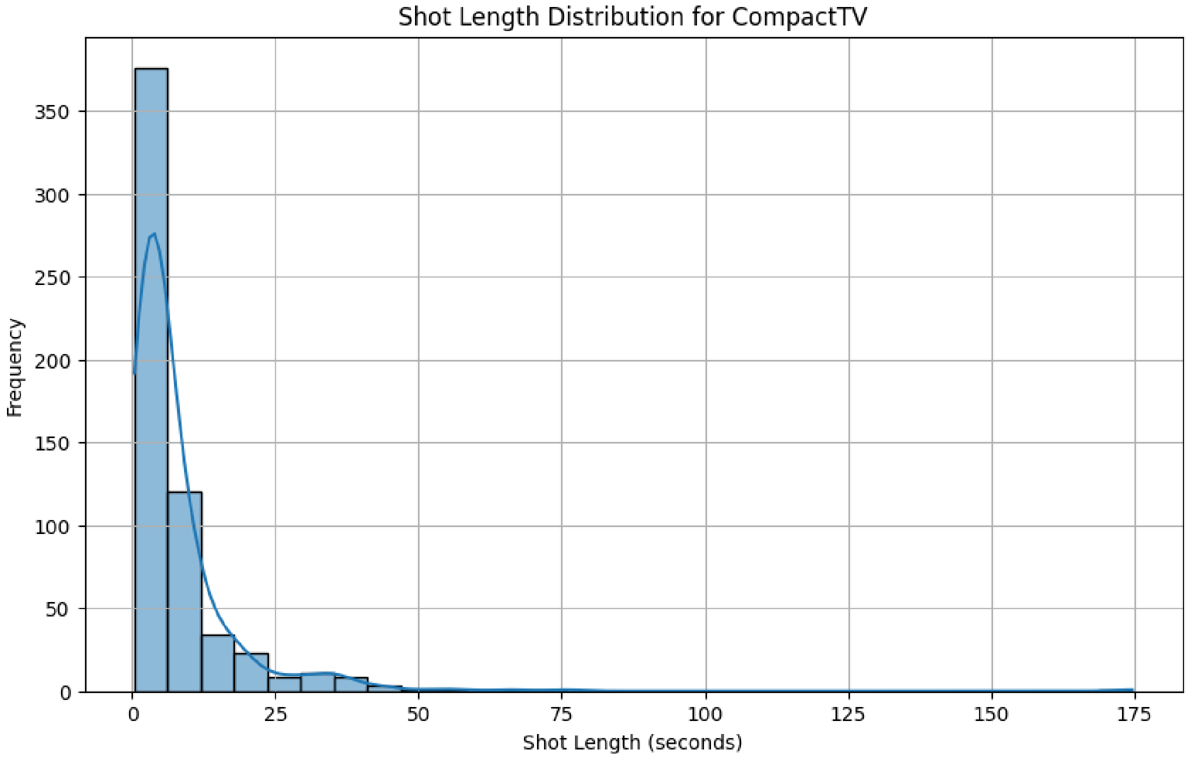


Figure A.4: Shot length distribution for CompactTV.

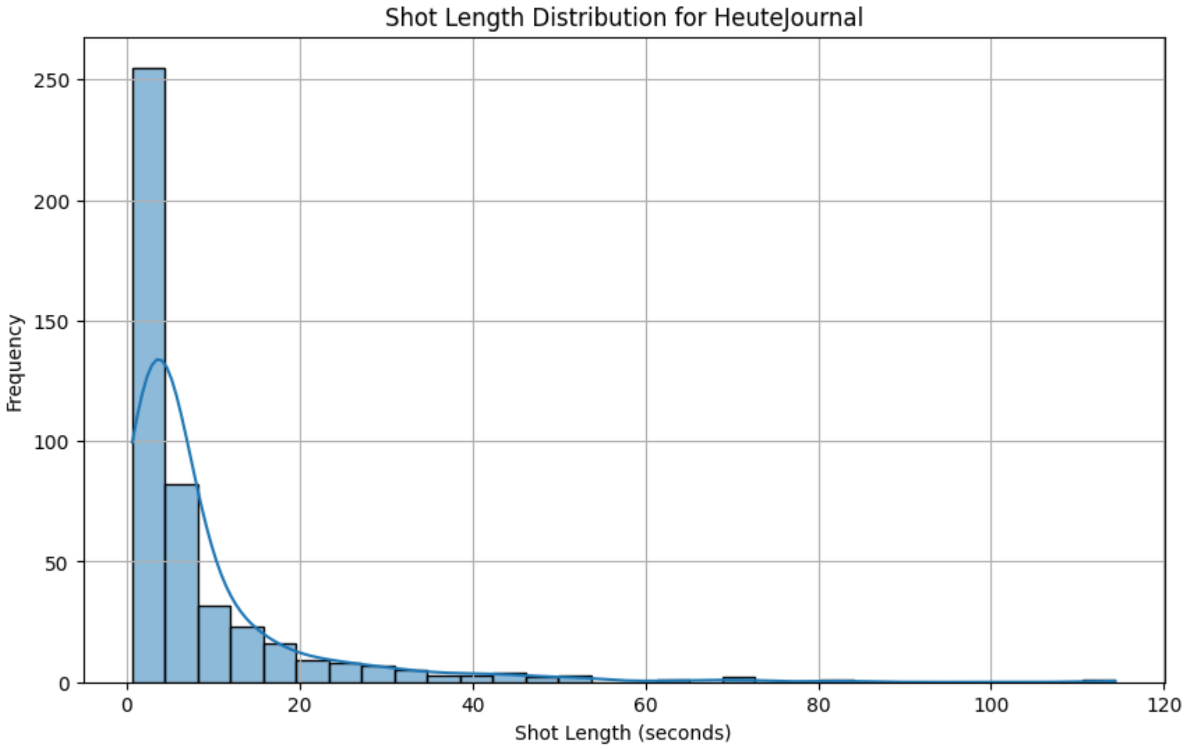
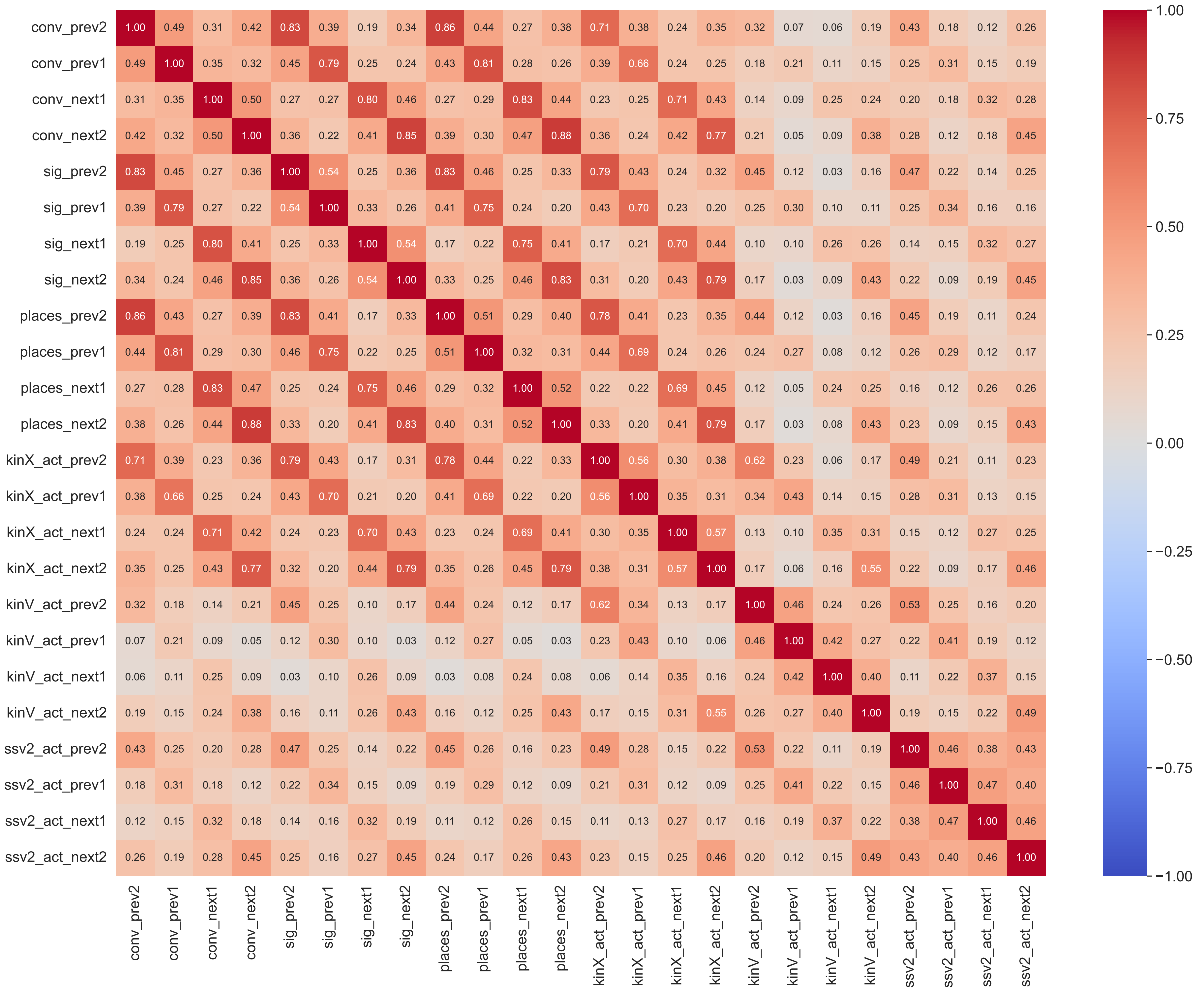


Figure A.5: Shot length distribution for HeuteJournal.

Table A.1: Pairwise Mann–Whitney U-Tests for shot lengths between channels. U-statistics are large due to the large sample size of 3012 shots.

Channel Pair	U Statistic	p-value	Significant?
Welt vs. Tagesschau	358,848.5	6.09×10^{-8}	Yes
Welt vs. BildTV	92,205.5	0.0438	Yes
Welt vs. CompactTV	199,207.5	7.28×10^{-5}	Yes
Welt vs. HeuteJournal	150,731.5	0.0041	Yes
Tagesschau vs. BildTV	125,927.5	5.63×10^{-14}	Yes
Tagesschau vs. CompactTV	296,958.5	0.4377	No
Tagesschau vs. HeuteJournal	222,850.5	0.0800	No
BildTV vs. CompactTV	118,166.0	4.37×10^{-7}	Yes
BildTV vs. HeuteJournal	91,962.5	1.26×10^{-6}	Yes
CompactTV vs. HeuteJournal	129,965.5	0.3641	No

A.4 Feature Correlations



	conv_prev2	conv_prev1	conv_next1	conv_next2	sig_prev2	sig_prev1	sig_next1	sig_next2	places_prev2	places_prev1	places_next1	places_next2	kinX_act_prev2	kinX_act_prev1	kinX_act_next1	kinX_act_next2	kinV_act_prev2	kinV_act_prev1	kinV_act_next1	kinV_act_next2	ssv2_act_prev2	ssv2_act_prev1	ssv2_act_next1	ssv2_act_next2
conv_prev2	1.00	0.49	0.31	0.42	0.83	0.39	0.19	0.34	0.86	0.44	0.27	0.38	0.71	0.38	0.24	0.35	0.32	0.07	0.06	0.19	0.43	0.18	0.12	0.26
conv_prev1	0.49	1.00	0.35	0.32	0.45	0.79	0.25	0.24	0.43	0.81	0.28	0.26	0.39	0.66	0.24	0.25	0.18	0.21	0.11	0.15	0.25	0.31	0.15	0.19
conv_next1	0.31	0.35	1.00	0.50	0.27	0.27	0.80	0.46	0.27	0.29	0.83	0.44	0.23	0.25	0.71	0.43	0.14	0.09	0.25	0.24	0.20	0.18	0.32	0.28
conv_next2	0.42	0.32	0.50	1.00	0.36	0.22	0.41	0.85	0.39	0.30	0.47	0.88	0.36	0.24	0.42	0.77	0.21	0.05	0.09	0.38	0.28	0.12	0.18	0.45
sig_prev2	0.83	0.45	0.27	0.36	1.00	0.54	0.25	0.36	0.83	0.46	0.25	0.33	0.79	0.43	0.24	0.32	0.45	0.12	0.03	0.16	0.47	0.22	0.14	0.25
sig_prev1	0.39	0.79	0.27	0.22	0.54	1.00	0.33	0.26	0.41	0.75	0.24	0.20	0.43	0.70	0.23	0.20	0.25	0.30	0.10	0.11	0.25	0.34	0.16	0.16
sig_next1	0.19	0.25	0.80	0.41	0.25	0.33	1.00	0.54	0.17	0.22	0.75	0.41	0.17	0.21	0.70	0.44	0.10	0.10	0.26	0.26	0.14	0.15	0.32	0.27
sig_next2	0.34	0.24	0.46	0.85	0.36	0.26	0.54	1.00	0.33	0.25	0.46	0.83	0.31	0.20	0.43	0.79	0.17	0.03	0.09	0.43	0.22	0.09	0.19	0.45
places_prev2	0.86	0.43	0.27	0.39	0.83	0.41	0.17	0.33	1.00	0.51	0.29	0.40	0.78	0.41	0.23	0.35	0.44	0.12	0.03	0.16	0.45	0.19	0.11	0.24
places_prev1	0.44	0.81	0.29	0.30	0.46	0.75	0.22	0.25	0.51	1.00	0.32	0.31	0.44	0.69	0.24	0.26	0.24	0.27	0.08	0.12	0.26	0.29	0.12	0.17
places_next1	0.27	0.28	0.83	0.47	0.25	0.24	0.75	0.46	0.29	0.32	1.00	0.52	0.22	0.22	0.69	0.45	0.12	0.05	0.24	0.25	0.16	0.12	0.26	0.26
places_next2	0.38	0.26	0.44	0.88	0.33	0.20	0.41	0.83	0.40	0.31	0.52	1.00	0.33	0.20	0.41	0.79	0.17	0.03	0.08	0.43	0.23	0.09	0.15	0.43
kinX_act_prev2	0.71	0.39	0.23	0.36	0.79	0.43	0.17	0.31	0.78	0.44	0.22	0.33	1.00	0.56	0.30	0.38	0.62	0.23	0.06	0.17	0.49	0.21	0.11	0.23
kinX_act_prev1	0.38	0.66	0.25	0.24	0.43	0.70	0.21	0.20	0.41	0.69	0.22	0.20	0.56	1.00	0.35	0.31	0.34	0.43	0.14	0.15	0.28	0.31	0.13	0.15
kinX_act_next1	0.24	0.24	0.71	0.42	0.24	0.23	0.70	0.43	0.23	0.24	0.69	0.41	0.30	0.35	1.00	0.57	0.13	0.10	0.35	0.31	0.15	0.12	0.27	0.25
kinX_act_next2	0.35	0.25	0.43	0.77	0.32	0.20	0.44	0.79	0.35	0.26	0.45	0.79	0.38	0.31	0.57	1.00	0.17	0.06	0.16	0.55	0.22	0.09	0.17	0.46
kinV_act_prev2	0.32	0.18	0.14	0.21	0.45	0.25	0.10	0.17	0.44	0.24	0.12	0.17	0.62	0.34	0.13	0.17	1.00	0.46	0.24	0.26	0.53	0.25	0.16	0.20
kinV_act_prev1	0.07	0.21	0.09	0.05	0.12	0.30	0.10	0.03	0.12	0.27	0.05	0.03	0.23	0.43	0.10	0.06	0.46	1.00	0.42	0.27	0.22	0.41	0.19	0.12
kinV_act_next1	0.06	0.11	0.25	0.09	0.03	0.10	0.26	0.09	0.03	0.08	0.24	0.08	0.06	0.14	0.35	0.16	0.24	0.42	1.00	0.40	0.11	0.22	0.37	0.15
kinV_act_next2	0.19	0.15	0.24	0.38	0.16	0.11	0.26	0.43	0.16	0.12	0.25	0.43	0.17	0.15	0.31	0.55	0.26	0.27	0.40	1.00	0.19	0.15	0.22	0.49
ssv2_act_prev2	0.43	0.25	0.20	0.28	0.47	0.25	0.14	0.22	0.45	0.26	0.16	0.23	0.49	0.28	0.15	0.22	0.53	0.22	0.11	0.19	1.00	0.46	0.38	0.43
ssv2_act_prev1	0.18	0.31	0.18	0.12	0.22	0.34	0.15	0.09	0.19	0.29	0.12	0.09	0.21	0.31	0.12	0.09	0.25	0.41	0.22	0.15	0.46	1.00	0.47	0.40
ssv2_act_next1	0.12	0.15	0.32	0.18	0.14	0.16	0.32	0.19	0.11	0.12	0.26	0.15	0.11	0.13	0.27	0.17	0.16	0.19	0.37	0.22	0.38	0.47	1.00	0.46
ssv2_act_next2	0.26	0.19	0.28	0.45	0.25	0.16	0.27	0.45	0.24	0.17	0.26	0.43	0.23	0.15	0.25	0.46	0.20	0.12	0.15	0.49	0.43	0.40	0.46	1.00

Figure A.6: Correlation Heatmap for Shot Similarity Features.

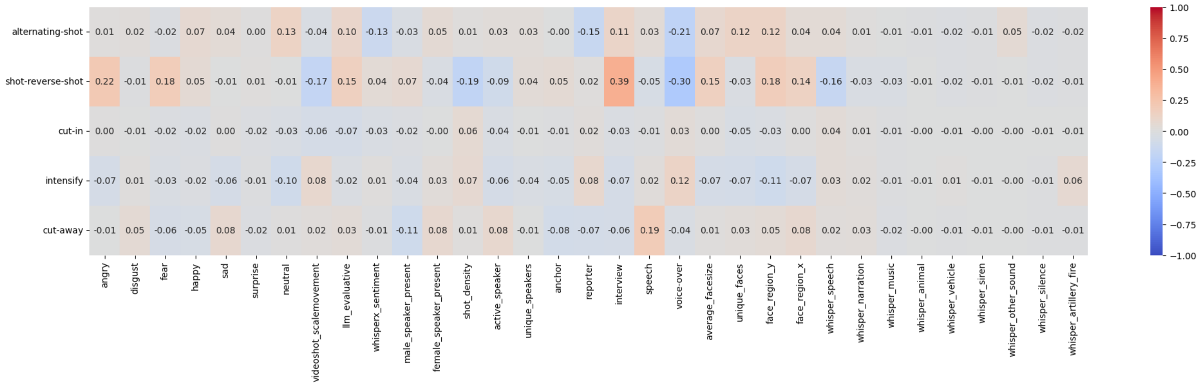


Figure A.7: Correlations between FEPs and remaining features.

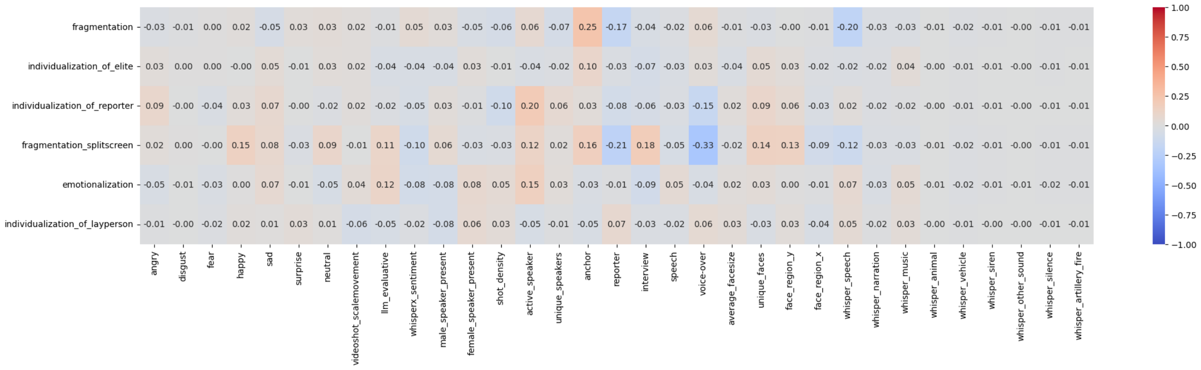


Figure A.8: Correlations between narrative strategies and remaining features.

A.5 Vision-Language Model and Classifier Comparison

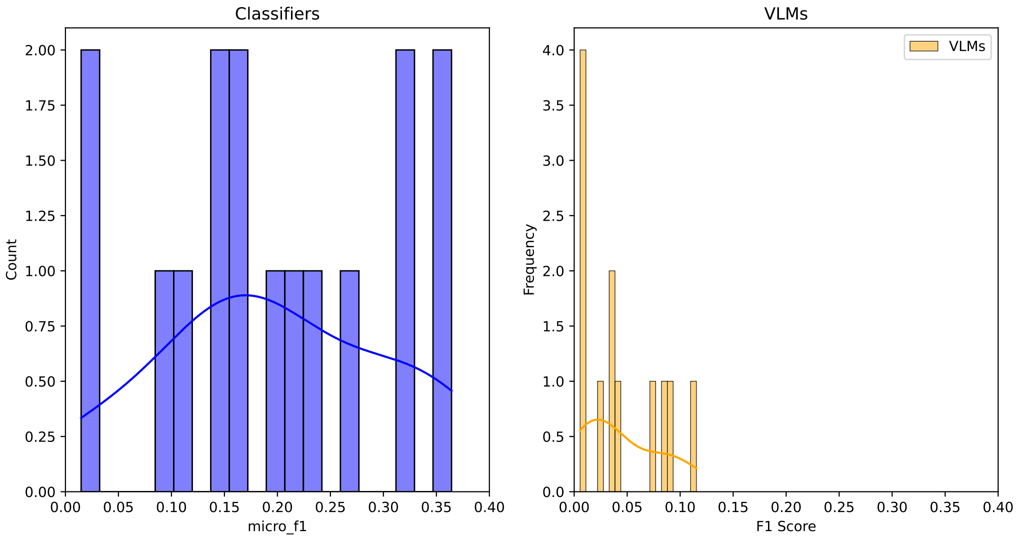


Figure A.9: Distribution of F1-scores by model type.

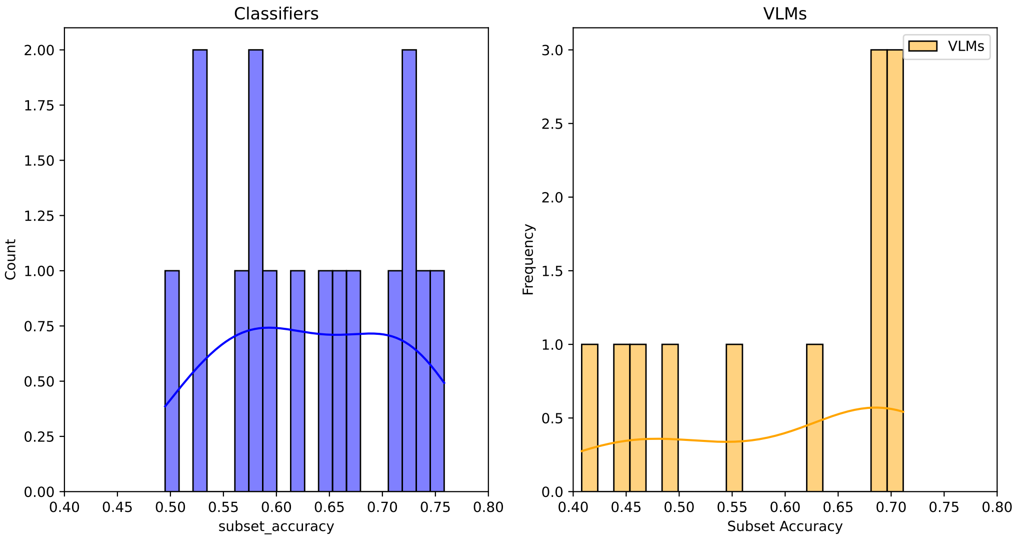


Figure A.10: Distribution of subset accuracy by model type.