Alma Mater Studiorum · Università di Bologna

DIPARTIMENTO DI INTERPRETAZIONE E TRADUZIONE
Corso di Laurea magistrale in Specialized Translation (classe LM-94)

TESI DI LAUREA
in COMPUTATIONAL LINGUISTICS

Decoding Medical Dramas: Identifying Isotopies through Multimodal Classification

CANDIDATA:
Alice Fedotova

RELATORE:
Alberto Barrón-Cedeño

CORRELATRICE:
Maja Miličević Petrović

Anno Accademico 2022/2023
Primo Appello

Acknowledgements

First of all, I would like to thank Prof. Alberto Barrón-Cedeño for always encouraging me to take on new challenges, especially at the beginning of this journey. I didn't expect that tinkering with a Raspberry Pi could transfer to my studies in any way. I would also like to thank Prof. Maja Miličević for the numerous discussions we had in my quest for finding connections between things. Thanks to both, and also to Prof. Adriano Ferraresi, for always being supportive throughout the whole TraTec process.

I would also like to thank Professors Guglielmo Pescatore, Marta Rocchi and Mirko Degli Esposti for involving me in their research on the analysis of medical dramas. This work wouldn't have been possible without all the efforts that went into the creation of the Medical Dramas Dataset.

Moreover, I wanted to thank my fellow colleagues from Lab 8. You all have truly kept me going. Thanks to Vicky for the powerful motivational speeches and for always cheering for me. Thanks to Paolo for convincing me to give VSCode another chance. Turns out I'm just one of those horrible people who prefer the light mode. Thanks to Hafsa for reminding me to take coffee breaks and for providing valuable feedback on some of the figures.

A special thanks to Roby for always being there and supporting me in everything I do. And especially for the *osake*.

Last but not least, thanks to my family for believing in me.

Abstract

The rise in processing power, combined with advancements in machine learning, has resulted in an increase in the use of computational methods for automated content analysis. Although human coding is more effective for handling complex variables at the core of media studies, audiovisual content is often understudied because analyzing it is difficult and time-consuming. The present work sets out to address this issue by experimenting with unimodal and multimodal transformer-based models in an attempt to automatically classify segments from the popular medical TV drama Grey's Anatomy into three narrative categories that are typical of the medical drama genre, also referred to as isotopies: the professional plot, the sentimental plot and the medical cases plot. To approach the task, this study explores two different classification approaches: the first approach is to employ a single multiclass classifier that directly predicts the target class labels, while the second involves using the one-vs-the-rest approach to decompose the multiclass task using a series of binary classifiers. We investigate both these approaches in unimodal and multimodal settings, with the aim of identifying the most effective combination of the two. The results of the experiments can be considered promising, given that the multiclass multimodal approach results in an F_1 score of 0.723, a noticeable improvement over the F_1 of 0.684 obtained by the one-vs-the-rest unimodal approach based on text. This provides support for the hypothesis that visual and textual modalities can complement each other and result in a better-performing model, which highlights the potential of multimodal approaches for narrative classification in the context of medical dramas. The main contributions of this dissertation are the following: (1) the creation of a multimodal corpus, containing keyframes and subtitles from 17 seasons of Grey's Anatomy, (2) an investigation into different task framing methods, namely a direct multiclass approach and a one-vs-the-rest approach, and (3) an extensive evaluation of various unimodal and multi-

modal transformer-based models, namely BERT, CLIP, and MMBT. The corpus and implementations are made available and pave the way for further research on automated content analysis in the context of medical dramas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the field of media studies, content analysis is an established methodology for the study of audiovisual products. A central aspect of content analysis is coding, which consists in assigning units of analysis to categories for the purpose of describing and quantifying phenomena of interest (Krippendorff, 1995). Previous research has identified three fundamental categories or "isotopies" that characterize the medical drama genre: the professional plot, the sentimental plot and the medical cases plot. In the context of medical dramas, content analysis can be conducted by assigning isotopies to segments, i.e. "portions of video characterized by space-time-action continuity" (Rocchi and Pescatore, 2022). This poses a challenge for automated approaches, as modern segmentation algorithms are not efficient at identifying units that are relevant for the identified isotopies. Additionally, coding requires trained annotators with a significant degree of expert knowledge and a good understanding of content analysis. Recognizing the complexity of the task and the need for more effective strategies, we experiment with different models to evaluate the possibility of streamlining the content analysis process for medical dramas. With this objective, we formulate the following:

> ***Hypothesis:*** Because subtitles and keyframes contain complementary information, a multimodal model that incorporates visual data in addition to text should perform better than a unimodal model trained exclusively on text.

To address this hypothesis, we aim to answer the following research questions:

Research Question 1: Is it better to approach the task with a single multiclass model or a one-vs-the-rest approach?

Research Question 2: Which modality is more informative for the task of predicting the isotopies?

Research Question 3: Does the inclusion of keyframes in addition to the subtitles result in higher performance as compared to only using the subtitles?

The objective of this study is to implement and evaluate unimodal and multimodal transformer-based models for the automatic identification of isotopies in the context of medical dramas. To achieve this, we first create a multimodal corpus by combining subtitles and keyframes extracted from 17 seasons of Grey's Anatomy, one of the longest-running medical drama series. Three models, namely CLIP, BERT, and MMBT, are trained using this corpus to explore the impact of different modalities on the identification of the isotopies. Additionally, we investigate two different approaches to the classification problem: a multiclass approach, which considers all isotopies simultaneously, and a one-vs-the-rest approach, which identifies one isotopy at the time. This study is organized into five core chapters:

Chapter 2 provides an overview of the foundational concepts at the basis of this study. It deals with the principles of natural language processing, the emergence of transformers, and their role in NLP. Then, it introduces computer vision and its intersection with NLP in the field of vision-and-language models, with a focus on multimodal fusion. The chapter closes with a discussion on the state of the art and a review of the relevant work in video understanding and narrative classification.

Chapter 3 outlines the essential data preprocessing steps required to prepare the data for the subsequent modeling process. It begins by explaining how subtitles are extracted, segmented, and labeled. The chapter further presents the process of discretization, which, despite leading to a loss in granularity, crucially reduces the complexity of the classification task. Lastly, the chapter provides a discussion on classification metrics, where accuracy, precision, recall, and F_1 score are introduced.

Chapter 4 presents the two different classification approaches that are investigated for isotopy identification: the direct multiclass approach and

the one-vs-the-rest approach. Various models, such as BERT for text, CLIP for vision, and MMBT for multimodal fusion, are described and their implementation details are provided. The results obtained from these models and approaches are then presented and discussed in relation to the research questions outlined at the beginning of the study.

Chapter 5 summarizes the main research outcomes. Notably, it was found that the multiclass multimodal approach, based on MMBT, obtained a significantly higher performance on the task of isotopy identification, outperforming unimodal models and reaching an F_1 score of 0.723. The one-vs-the-rest approach generally proved to be more effective for unimodal models, while multiclass MMBT surpassed its one-vs-the-rest counterpart, suggesting that visual data might help MMBT to disambiguate instances more effectively in the multiclass approach. Analysis of the unimodal models revealed the text was more informative than keyframes for the task of predicting the isotopies. A few limitations are also discussed, along with some promising areas for future research.

Chapter 2

Background

This chapter introduces the fundamental ideas and techniques at the basis of this research. Section 2.1 provides an introduction to natural language processing and the deep learning concepts required to understand the current approaches in the field. Following this, computer vision is introduced in Section 2.2, a field that has also been heavily influenced by deep learning. Building on these individual modalities, we then look at the intersection of vision and language in Section 2.3, which covers the emerging field of multimodal vision-and-language models. Lastly, we present an overview of related work on video understanding in Section 2.4.

2.1 Natural Language Processing

One of the main fields that this dissertation is based on is natural language processing, which can be defined as follows:

> *Natural language processing is an area of research in computer science and artificial intelligence (AI) concerned with processing natural languages such as English or Mandarin. This processing generally involves translating natural language into data (numbers) that a computer can use to learn about the world* (Lane et al., 2019, p. 4).

As natural languages evolve over time and are difficult to describe precisely using explicit rules, one of the key problems in NLP is how to translate

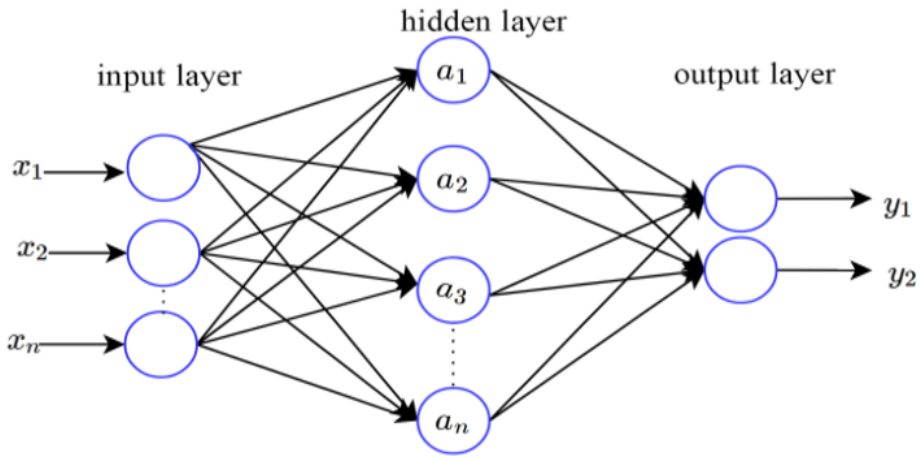


Figure 2.1: A simple feed-forward neural network (Hodo et al., 2017).

natural language into a numerical representation that can be processed by a computer (Lane et al., 2019, p. 4). In the past, NLP relied on handcrafted rules and linguistic patterns to achieve this goal. However, advances in machine learning resulted in a paradigm shift towards statistical NLP, which consists in training a model to learn directly from large scale datasets instead of relying on explicitly programmed rules (Lane et al., 2019, p. 63). Current state-of-the-art models in NLP are based on deep learning, which is a subset of machine learning that leverages multi-layered artificial neural networks to extract patterns from raw input data.

In the context of neural networks, learning involves finding a set of values for the weights in the layers of a network, such that the network will correctly associate inputs to some specific output (Chollet, 2020, p. 11). Neural networks typically function as supervised learning models. Supervised learning requires labelled training data, where both inputs and their corresponding desired outputs, known as targets, are provided to the model (Lane et al., 2019, p. 185). For example, in a supervised learning task for email classification, each email input would be labeled as either "spam" or "ham". A representation of a simple feed-forward neural network is illustrated in Figure 2.1. This feed-forward neural network comprises an input layer with n neurons (x_1, x_2, x_n), a hidden layer with n neurons $(a_1, a_2, ..., a_n)$, and an output layer with two neurons (y_1, y_2). In the initial stage of training a feed-forward neural network, each connection between neurons is given a random weight,

denoted as w_{ij}, where i corresponds to an input neuron $(x_1, x_2, ..., x_i)$ and j represents a neuron in the hidden layer $(a_1, a_2, ..., a_j)$. These weights are first multiplied and summed to compute a single value in each neuron of the hidden layer:

$$z_j = \sum_{i=1}^{n} w_{ij} x_i, \tag{2.1}$$

Then, the weighted sum z_j is passed to an activation function such as the sigmoid or the Rectified Linear Unit (ReLU). This introduces non-linearity into the model, enabling it to learn complex patterns from the data. Activation functions essentially decide whether the neuron should 'fire' or not based on the input. If the input surpasses a certain threshold, the neuron gets activated and consequently passes its result to the next layer; otherwise, it remains inactive (Chollet, 2020, p. 72). The same process is then repeated in the output layer, where the choice of the activation function can vary based on the nature of the problem at hand. In a binary classification task, the sigmoid function is typically chosen due to its property of producing outputs between 0 and 1, well-suited for binary outcomes. For multiclass classification tasks, the softmax function is generally preferred as it outputs a probability distribution across multiple classes (Chollet, 2020, p. 114).

During the training phase of a neural network, a feedback signal is employed to adjust the weights based on the output error of the network. This error, known as the loss, is calculated using a loss function, which quantifies how far off our predictions are from the actual targets. Common examples of loss functions include Mean Squared Error for continuous targets in regression problems, and Cross Entropy for categorical targets in classification. The process of propagating this error back through the network to update the weights is known as backpropagation, and the specific algorithm that dictates how the weights should be adjusted in order to minimize the loss is referred to as the optimizer. Some frequently used optimizers include Stochastic Gradient Descent, RMSprop, and Adam. This systematic adjustment of weights, based on the computed error, is what allows the network to 'learn' from its inputs and gradually improve its predictions (Chollet, 2020, p. 29).

The process of training a neural network also depends on predetermined control variables called hyperparameters, such as epochs, batch size, and learning rate. Unlike parameters, which in the case of neural networks are the weights that the model learns during training, hyperparameters are set

before the training process begins. Epochs indicate the number of times that the entire training dataset is processed by the network, while batch size refers to the number of instances from the training set that are processed by the network before updating its weights. The learning rate, on the other hand, defines the magnitude of the adjustments to the weights. Hyperparameters play a critical role in managing the balance between overfitting and underfitting. Overfitting occurs when the model is too complex and fits the training data too closely, while underfitting happens when the model is too simple and cannot capture the patterns in the data. For example, if a model is trained for too many epochs, its performance might start to degrade due to overfitting (Chollet, 2020, p. 76).

2.1.1 Transformer Models

Expanding on the topic of deep learning in NLP, we will now transition to one of its most notable applications: the transformer model, introduced by Vaswani et al. (2017) in their paper, "Attention is All You Need". The emergence of the transformer marked a significant advancement in the field of NLP, surpassing former state-of-the-art architectures such as long short-term memory networks (LSTMs). The success of LSTMs had to do with their unique memory mechanism, which involves learning what information to retain and what to discard while processing sequences word by word. While this sequential, word by word processing makes LSTMs effective in NLP tasks due to their ability to remember past information and capture word order, their memory is generally limited to recent inputs, which makes it challenging for LSTMs to process long-distance dependencies in language (Tunstall et al., 2022).

On the other hand, the transformer processes the entire sequence at once. This enables it to efficiently capture global dependencies and relationships between words, thanks to a mechanism called self-attention. Self-attention is part of the encoder component of a transformer, which is composed of several encoder blocks (Figure 2.2) that first convert an input sequence of tokens into a sequence of embedding vectors, i.e. into numerical representations that can be processed by the network (Tunstall et al., 2022). The concept of self-attention further extends to multi-head attention, which means that multiple attention patterns can be computed simultaneously, providing a comprehensive view of the sequence's context (Figure 2.3). Having several

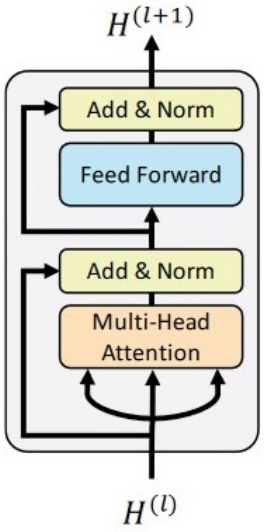


Figure 2.2: Transformer encoder block. Adapted from (Lu et al., 2019).

heads allows the model to focus on different aspects at once. For instance, one head may focus on subject-verb interaction, while another looks for nearby adjectives (Ryu and Lewis, 2021).

The multi-head attention output in a transformer encoder block is then normalized and processed through a simple feed-forward neural network. This cycle, repeated in multiple encoder blocks, refines the input into increasingly sophisticated representations[1]. The final output of the encoder are contextualized embeddings, representations that carry the nuanced semantic information about each word while considering its context within the sequence. Given that transformers are not sequential like LSTMs, they do not inherently understand the order of words in a sequence. To address this, transformers make use of positional encodings, which are added at the beginning to the input embeddings, and function like unique identifiers that keep track of the original order of words (Tunstall et al., 2022).

[1]Research suggests that this process mirrors the traditional NLP pipeline, with the encoder initially learning simpler linguistic concepts like part-of-speech tagging and parsing, and then more complex aspects such as named entity recognition, semantic roles, and coreference (Tenney et al., 2019)

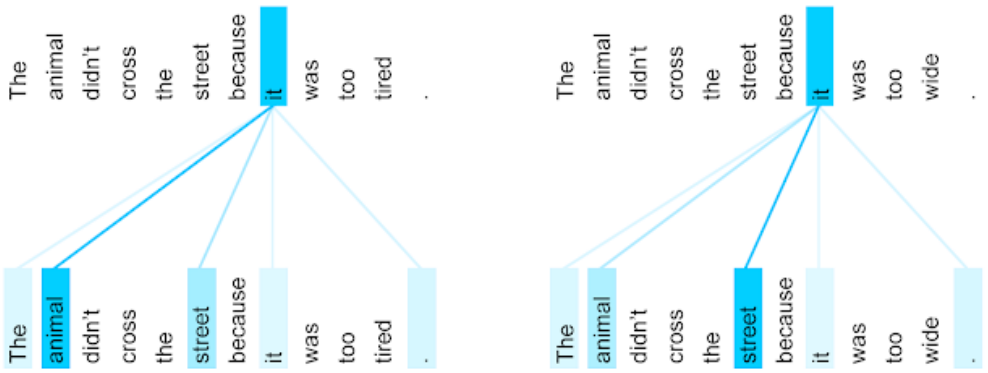


Figure 2.3: Self-attention distribution for the word "it", from `https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html`.

2.1.2 Encoder Models

Initially, the transformer architecture was introduced for machine translation (Devlin et al., 2019). In the full transformer model, the encoder reads and processes the entire input sequence and the decoder takes these representations to generate translations in other languages. However, many NLP tasks do not require a decoder. This gave rise to encoder models, which are used to generate contextual embeddings of the input text that can be used for a wide range of tasks. An example of an encoder model is BERT (Bidirectional Encoder Representations from Transformers), introduced by Devlin et al. (2019). One of the main features of BERT is the idea of bidirectional pre-training. During its pre-training process, BERT uses a method called masked language modeling, where it randomly masks words in the sequence and then predicts those masked words based on the context provided by the non-masked words – those that come before and after the masked word in the sequence. This, along with next sentence prediction, during which the model learns to predict whether a given sentence logically follows a preceding one, enables BERT to learn meaningful representations from large text corpora. When using BERT, a model that has been pre-trained on these tasks is typically used as a starting point and then fine-tuned for the task at hand, i.e. further trained on a small, task-specific dataset (Tunstall et al., 2022).

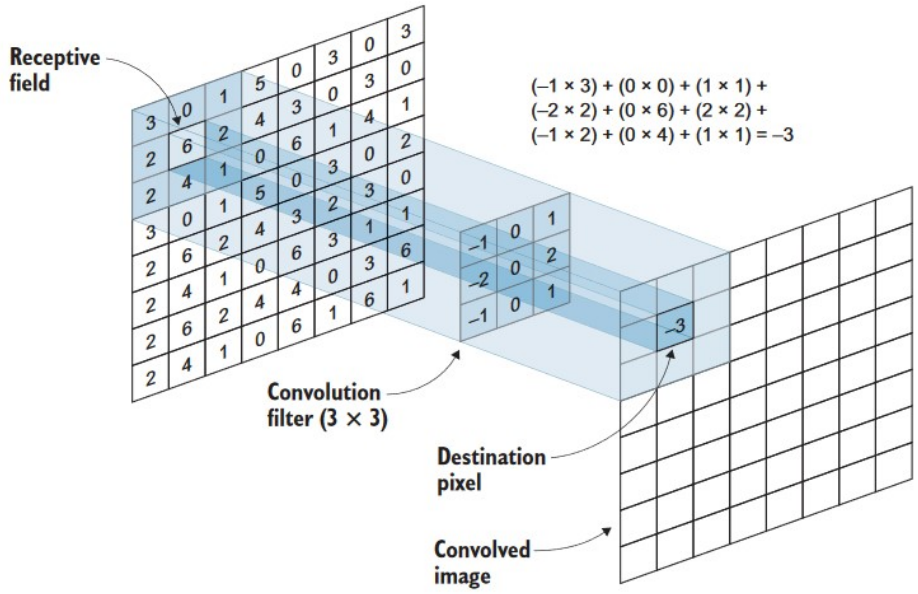


Figure 2.4: A 3×3 filter sliding over the input image. In a grayscale image, each pixel is represented by a number which corresponds to the brightness of the pixel (Elgendy, 2020).

2.2 Computer Vision

Computer vision (CV) is a subfield of machine learning that deals with enabling computers to process visual data. The foundation of computer vision tasks is feature extraction, which consists in transforming an input image into a complex representation emphasizing various image characteristics at different network layers. Convolutional neural networks (CNNs) are the most notable deep learning models in computer vision (Guo et al., 2016). The main component of a CNN is the convolutional layer. This layer applies a series of filters to the input image, each capturing some specific feature. A filter is a small matrix of weights, typically of size 3x3, which is applied to the image using a process called convolution. Starting from the top-left corner of the image, the filter slides across the image, moving right and down by a set number of pixels (referred to as the stride) at each step. At every position, the filter's values are multiplied with the corresponding pixel values beneath it in the image, and these products are summed up (Figure 2.4). Initially,

the values within the filters are set randomly. By adjusting the weights in the filters over many rounds of training, the network learns which features of the image are more informative for the task at hand (Elgendy, 2020).

It is important to note that a CNN does not go from the image input to the features directly in one layer. The feature-learning process happens step by step in tens or hundreds of layers. It is recognized that network depth is of crucial importance in CV, with very deep models often leading the results on computer vision benchmarks (He et al., 2016). The ResNet (Residual Network) architecture, proposed by Microsoft researchers led by He et al. (2016), constituted a significant advancement in this sense. With ResNet, models could be designed with an unprecedented number of layers, with some versions even reaching up to 152, thereby improving the potential for advanced performance in CV tasks (He et al., 2016). Pre-training is also a widespread technique in computer vision. Since the publication of large datasets such as ImageNet (Deng et al., 2009), many architectures have been trained on them and their weights made publicly available to be used for transfer learning. Building upon ResNet, OpenAI developed CLIP (Radford et al., 2021), which provides a pretrained, modified version of ResNet-50, named RN50x4, as one of its visual encoders. The potential of CLIP, particularly with the RN50x4 encoder, was underscored by a study conducted by Shen et al. (2021), which examined the performance of vision-and-language models (cf. Section 2.3) when the visual encoder is switched to CLIP. Their findings indicated that CLIP's RN50x4 surpassed the conventional ResNet-152 feature extractor, which is a common visual extractor in many vision-and-language models.

2.3 Vision and Language

Driven by the success of pre-trained models in the fields of natural language processing and computer vision, more and more research began to focus on multimodal tasks (Wang et al., 2022). In the field of multimodal deep learning, a modality can be defined as "a particular way or mechanism of encoding information" (Guo et al., 2019). Multimodal data can be extracted from various sources, such as text, images, audio, and video. As multimodal data often represents an object from different viewpoints, which can be complementary in contents, it can potentially be more informative than unimodal data. However, there are also instances where the modalities end up compet-

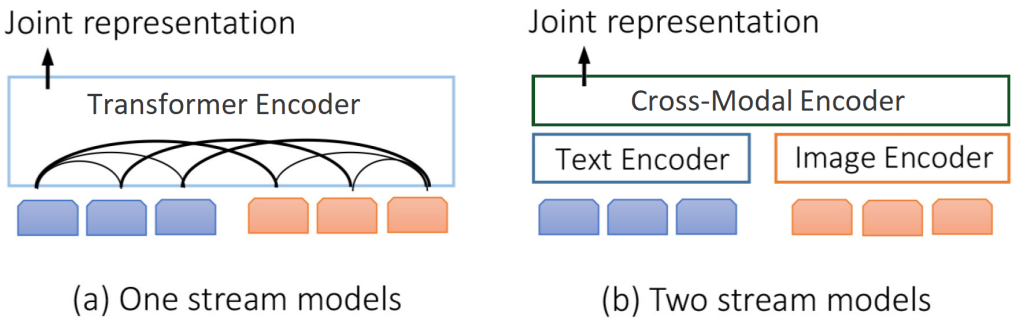
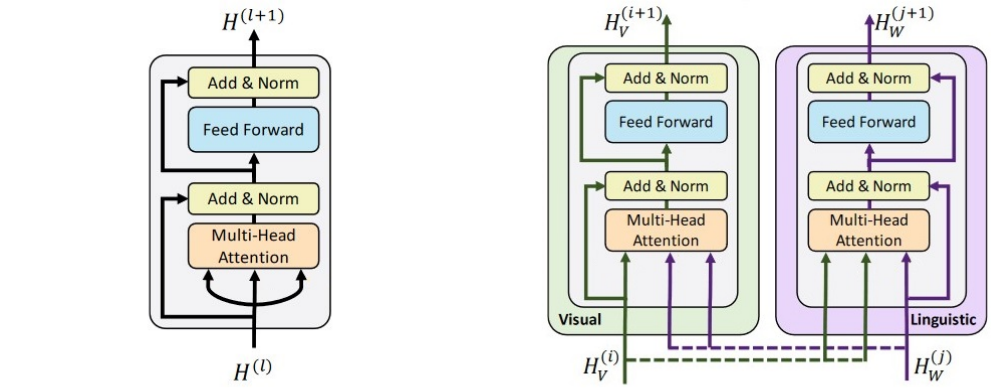


Figure 2.5: One-stream and dual-stream models. Adapted from Zhang et al. (2022).

ing with each other, causing multimodal models to underperform compared to the unimodal ones (Huang et al., 2021). Models that leverage both visual and textual information are known as vision-and-language (VL) models. The emergence of transformers in NLP has greatly influenced vision-and-language models, resulting in a multitude of models that extend BERT (Devlin et al., 2019) to learn multimodal representations (Bugliarello et al., 2021). One challenge in multimodal tasks is multimodal fusion, which involves integrating information from multiple modalities. In multimodal fusion, information is typically fused at three levels: input (early fusion), intermediate representation (mid fusion), and prediction (late fusion). However, late fusion is less commonly used with multimodal transformers due to the advantages of learning stronger joint representations across modalities (Xu et al., 2023). Hence, we will focus on early fusion and mid fusion[2].

Common early-fusion-based multimodal transformers are one-stream models, also known as single-stream models (Xu et al., 2023). Some examples include MMBT (Kiela et al., 2019) and VisualBERT (Li et al., 2019). One-stream models allow the adoption of the merits of BERT with only minimal modifications to its architecture. Single-stream architectures assume that the potential correlation and alignment between the modalities is simple, and that it can be learned by a single transformer encoder (Figure 2.5a). The text embeddings and the image features are usually concatenated together,

[2]Late fusion consists in using two separate unimodal models and averaging their predictions at the end (Baltrušaitis et al., 2018).



(a) Standard encoder transformer block (b) Co-attention transformer layer

Figure 2.6: Co-attention in ViLBERT. Adapted from (Lu et al., 2019).

adding some special embeddings to indicate positions and modalities, and then fed into a transformer encoder. As the single-stream structure applies self-attention directly on the two modalities, there is a chance it may overlook interactions within the same modality. Therefore, certain studies suggest the use of a dual-stream architecture to better capture the interaction between visual and linguistic elements (Du et al., 2022).

Unlike single-stream architectures, dual-stream or two-stream architectures employ a cross-modal encoder, also referred to as a cross-attention or co-attention layer depending on the specific model (Figure 2.6). Examples include LXMERT (Tan and Bansal, 2019) and ViLBERT (Lu et al., 2019). Dual-stream architectures use two different encoders, one for processing textual inputs and the other for handling image data, each applying intra-modal self-attention to extract information from their respective modalities. A cross-modal encoder is then added on top of this two separate modules to exchange the information between the two modalities (Figure 2.5b). Co-attention is similar to the self-attention mechanism in traditional transformers, but it differs in that it allows the model to dynamically focus on relevant parts of both modalities simultaneously, learning which words or phrases are most relevant to the parts of the image, and vice versa. Two-stream architectures are an example of mid fusion, as they feature separate processing paths for different modalities and merge the resulting information at a midpoint in the model's structure (Rayavarapu et al., 2019).

There are, however, also VL models that do not fall into either of these

categories. One of such models is CLIP (Contrastive Language-Image Pre-Training), which processes image and text data separately through two distinct encoders, and then maps them into a common semantic space through a straightforward method such as a dot product instead of a complex transformer network (Radford et al., 2021). CLIP focuses on aligning texts and images at a semantic level, which makes this model particularly suitable for tasks like zero-shot classification where the model is asked to classify data into classes it has not explicitly seen during training (Radford et al., 2021). The interaction between the modalities in CLIP is relatively shallow, primarily focusing on semantic alignment rather than a deep understanding of the interplay between the visual and the linguistic data (Du et al., 2022).

2.4 Related Work

With its ability to engage several human faculties at once, audiovisual content is able to present information in a more multifaceted way compared to static images or text. However, the addition of the time element through shots and scenes makes it extremely complex for machine learning models to understand the content of a video. One of the biggest challenges in the fields of natural language processing and computer vision is developing the ability for machines to analyze and summarize the narratives conveyed in videos, making them more searchable and accessible (Tapaswi, 2016). Despite the fact that there is yet no agreement on which modality is best when eliciting high-level meaning from audiovisual content, researchers believe that two or more modalities are better than one (Bayoudh et al., 2022).

Compared to visual and auditory information, textual clues are less explored for video understanding (Weng et al., 2021). In the broader context of movies and TV shows, speech may sometimes be correlated with the action (e.g., "Raise your glasses to..."), but it is more frequent for it to be completely uncorrelated (Nagrani et al., 2020). In the field of sentiment analysis, related work has been conducted on the TV show Friends. Zahiri and Jinho (2017) employ a CNN architecture with word2vec embeddings for the purpose of detecting emotions from written dialogue, obtaining accuracies of 37.9% and 54% for fine- and coarse-grained emotions respectively. They observe that emotions are not necessarily conveyed in the text, and that disfluencies, metaphors, and humor make the task particularly challenging.

Vision-and-language approaches to video understanding can be divided into two types: one with image (cf. Section 2.3), and another with video (Sun et al., 2019; Zhu and Yang, 2020). In the context of video, this usually consists in densely extracting multiple frames, as it is reasonable to assume that training an effective video-and-language model requires lots of samples from the video channel (Lei et al., 2022). As demonstrated by Li et al. (2021), leveraging both video and subtitles achieves the best performance on the VALUE benchmark, which includes 11 video understanding tasks from a variety of datasets and video genres. A similar result is reported by Liu et al. (2020) on the task of video-and-language inference, which consists in analyzing a video clip paired with a natural language hypothesis and determining whether the hypothesis is supported or contradicted by the information conveyed in the video.

However, it is actually an open question whether training a model using multiple frames is beneficial for downstream tasks, and if so, whether the gains in performance justify the significant increase in computational costs (Lei et al., 2022). Despite the fact that most video-and-language models are typically trained using multiple video frames, some studies suggest that strong performance on challenging benchmarks can be achieved using just a single frame (Lei et al., 2022; Buch et al., 2022). Furthermore, the difficulty of making recognition decisions is intrinsically linked to the type of category being classified. For instance, recognizing static subjects like dogs and cats, or sceneries such as forests or seas, may only require a single frame. However, distinguishing nuanced actions, such as "drinking coffee" versus "drinking beer", often requires more frames (Wu et al., 2019).

As for single-frame approaches, Lei et al. (2022) introduced a model called Singularity. During fine-tuning, a random single frame is used for training, while multiple uniformly sampled frames from videos are used to output video-level predictions during inference. Even if trained with less data, Singularity excels on video retrieval and captioning benchmarks such as DiDeMo and ActivityNet Captions, proving to be highly efficient in both training time and memory usage. On the other hand, Buch et al. (2022) proposed a model called Atemporal Probe (ATP) that processes videos by focusing on one selected frame at the time. The choice of the frame is driven by an algorithm within the ATP that identifies and selects the most informative frame for understanding the video content. This approach performs surprisingly well on multiple benchmarks including MSR-VTT-MC (Xu et al., 2016), a large-scale

dataset for video captioning, even against more complex models.

To the best of our knowledge, this is the first work on narrative classification for the medical drama genre. In the context of cinema, a similar work is the Movie Narrative Dataset (MND), introduced by Liu et al. (2023). MND consists of 6,448 annotated scenes from 45 movies, manually labeled by multiple annotators into 15 key story elements. To benchmark the task of classifying scenes based on their narrative function, the authors of MND utilized an XGBoost classifier trained on temporal features and character co-occurrence patterns. With five-fold cross-validation, the classifier obtained an F_1 score of 0.31, which, while still leaving room for improvement, is statistically significant and outperforms a static baseline classifier.

In the field of multimodal misogyny identification, Muti et al. (2022) proposed a multimodal approach for detecting misogynistic content in memes. Their approach is based on a multimodal bi-transformer model (MMBT), using early fusion to combine textual and visual embeddings. Their approach was evaluated on the MAMI shared task (Fersini et al., 2022), obtaining macro-averaged $F_1 = 0.727$ in Task A (binary misogyny identification) and weighted $F_1 = 0.710$ in Task B (multi-label classification into four potentially co-occurring categories). This results demonstrate the effectiveness of a multimodal approach for identifying misogynistic content in memes.

As for the methodology, the present dissertation draws upon the approach proposed by Muti et al. (2022). Given its success in multimodal misogyny identification, we investigate whether MMBT can achieve similar results when it comes to isotopy identification, which collocates this work in the context of image-based approaches to video classification. This choice is also motivated by studies showing the potential of using only a single frame such as the ones by Lei et al. (2022) and Buch et al. (2022), which demonstrate that strong performance can be achieved without considering multiple frames. Additionally, the decision to consider a single frame is influenced by the substantial increase in computational costs associated with analyzing multiple frames (Lei et al., 2022), which indeed presents significant challenges in terms of resource requirements and processing time.

Chapter 3

Evaluation Framework

3.1 Task Definition

In this section, we present an outline of the problem at the core of this thesis, clarify the main terms and concepts, and introduce the hypothesis and research questions driving the study.

The primary goal of this dissertation is to classify segments from the TV series Grey's Anatomy into three distinct isotopies: the sentimental plot, the professional plot, and the medical case. The classification is approached as a multiclass problem, employing both unimodal and multimodal models.

The term 'segment' refers to a unit of the audiovisual product characterized by continuity in terms of space, time, and action, as well as consistency in terms of thematic and narrative elements (Rocchi and Pescatore, 2022). The term 'isotopy', roughly equivalent to the term 'plot', refers to a recurring pattern of narrative features that are specific to a given genre – in this case, the medical drama (Pescatore and Rocchi, 2019).

The proposed hypothesis is as follows:

> ***Hypothesis:*** Because subtitles and keyframes contain complementary information, a multimodal model that incorporates visual data in addition to text should perform better than a unimodal model trained exclusively on text.

To investigate this hypothesis, we aim to develop a multimodal corpus by expanding an existing dataset of segments annotated with start and end timestamps, as well as the corresponding isotopies. Textual features would be extracted by temporally aligning the subtitles with the segments, while visual features would be obtained by extracting a frame between the start and the end of each segment, which we will refer to as 'keyframe'. This would allow us to experiment with unimodal and multimodal models for the automatic identification of isotopies, which can streamline the process of content analysis by automatically detecting narrative patterns in medical dramas. In order to begin addressing the proposed hypothesis, we aim to answer the following research questions:

> ***Research Question 1:*** Is it better to approach the task with a single multiclass model or a one-vs-the-rest approach?
>
> ***Research Question 2:*** Which modality is more informative for the task of predicting the isotopies?
>
> ***Research Question 3:*** Does the inclusion of keyframes in addition to the subtitles result in higher performance as compared to only using the subtitles?

To answer these research questions, we propose the following evaluation framework:

1. Extract the textual features from the subtitles and the visual features from the keyframes, based on the provided temporal annotation.

2. Process the data into a suitable format for classification.

3. Train and evaluate unimodal models (first using only visual features, then using only textual features) for the multiclass classification task, implementing both multiclass and one-vs-the-rest approaches.

4. Train and evaluate multimodal models (using both visual and textual features) for the multiclass classification task, also implementing both multiclass and one-vs-the-rest approaches.

3.2 Corpus Creation

3.2.1 Background

The present work builds upon the Medical Dramas Dataset outlined in Rocchi and Pescatore (2022). The dataset includes more than 400 hours of video and consists of eight US medical dramas, for a total of 32 seasons and 608 episodes[1]. Isotopy assignment, also referred to as 'coding', was conducted according to a three-step content analysis protocol (Rocchi and Pescatore, 2022). First, three isotopies underlying the medical drama genre were identified: the medical cases plot, the professional plot, and the sentimental plot. According to Pescatore and Rocchi (2019), the isotopies can be defined as follows:

The medical cases plot (MC) is related to the storylines that usually change between each episode, introducing new narrative elements and a variety of characters into the hospital setting.

The professional plot (PP) deals with the relationships and dynamics within the hospital among doctors and other medical staff.

Lastly, the sentimental plot (SP) comprises the emotional and personal relationships between the main characters throughout the series. It covers a wide sphere of emotions such as friendship, love, empathy, and conflict.

The second step involved breaking down each episode into segments, which are defined as the units of the audiovisual product that possess continuity in terms of space, time, and action, as well as consistency in terms of thematic and narrative elements (Rocchi and Pescatore, 2022). For each segment, start and end times were identified and recorded. This aspect is especially important, as it allowed the subsequent alignment with the text of the subtitles (see Section 3.2.2).

The actual coding phase followed the identification of the segments. During this phase, the appropriate isotopies were assigned to each previously identified segment, taking into account their development over time and not treating them as independent segments. A weight from 0 to 6 was assigned

[1]The dataset is available at `https://osf.io/24tus/`.

code	segm_start	segm_end	time	pp	sp	mc
GAS13E01	00:00:00	00:00:44	00:00:44	NA	NA	NA
GAS13E01	00:00:44	00:00:49	00:00:05	NA	NA	NA
GAS13E01	00:00:49	00:02:18	00:01:29	0	6	0
GAS13E01	00:02:18	00:02:36	00:00:18	0	2	4
GAS13E01	00:02:36	00:03:18	00:00:42	0	6	0

Table 3.1: Snapshot of the original Medical Dramas Dataset.

to each of the plots. If a segment could only be attributed to a single plot, a weight of 6 was assigned to that plot and a weight of 0 to the other two. When there were overlaps between narrative lines, a weight was assigned to each of the co-occurring narratives according to their relevance in the segment. In some cases, segments were not attributable to either of the isotopies and all three were marked as "NA" (Rocchi and Pescatore, 2022).

An example showing some instances from the Medical Dramas Dataset is illustrated in Table 3.1, where each row corresponds to a segment. The code "GA" refers to the TV show Grey's Anatomy, and it is followed by the season ("S13") and the episode ("E01"). The columns "segm_start" and "segm_end" are the start and end timestamps respectively, whereas "time" indicates the duration of the segment. Lastly, "pp", "sp", and "mc" contain the distribution of the isotopies. Segments labeled as "NA" were not attributable to either of the isotopies.

3.2.2 Data Extraction

The availability of start times and end times for each segment allowed for the alignment of the dataset with another source of data tagged with temporal information: the subtitle track of the episodes. Each subtitle has four parts in a SubRip Subtitle (SRT) file[2]: a counter indicating the number of the subtitle; start and end timestamps; one or more lines of text; and an empty line indicating the end of the subtitle. By relying on these features, the SRT files were processed to extract the timestamps and the text of the subtitles.

[2] `https://docs.fileformat.com/video/srt/`

segm_start	segm_end	sub_average	sub_start	sub_end
00:06:13	00:06:20	00:06:18.594	00:06:17.587	00:06:19.602
00:06:20	00:07:14	00:06:20.179	00:06:19.635	00:06:20.723
00:06:20	00:07:14	00:06:21.874	00:06:20.755	00:06:22.994

Table 3.2: The subtitle in line 2 starts slightly before the segment to which it was assigned. Nevertheless, the average (00:06:20.179) is comprised between the start (00:06:20) and the end of the segment (00:07:14). Therefore, this segment contains the majority portion of the subtitle.

For the purpose of aligning the subtitles with the Medical Dramas Dataset, a method for assigning each of the subtitles to the corresponding segment was identified. Inspired by Tapaswi et al. (2014), in which subtitles occurring at video shot boundaries were assigned to the shot which has a majority portion of the subtitle, the average of each subtitle's timespan was used as the criterion for the alignment. For example, given a subtitle that starts at 00:00:00.804 and ends at 00:00:02.701, the average is 00:00:01.752. If a segment starts at 00:00:00.000 and ends at 00:00:07.000, then the subtitle is part of that segment. By doing so, a subtitle that overlaps with two different segments is assigned to the one where it appears on the screen for the longest amount of time. Table 3.2 illustrates this approach more in detail.

In addition to processing the subtitles, keyframes were also extracted from each of the episodes. A script based on OpenCV[3], an open-source computer vision library, was developed to accomplish this task. For each video, the midpoint of each segment was calculated based on the start and end times of the segment. The corresponding keyframe is then extracted and saved as a JPG file. Table 3.3 illustrates a few examples from the corpus, consisting of different segments and their timestamps, as well as the text obtained from the subtitles, the filenames of the keyframes and the assigned isotopies. Unique IDs were also assigned in column "id". In contrast with Table 3.1, the segments up to 00:00:49 are missing because they were labeled as NA. An example of a keyframe is also shown in Figure 3.1.

[3] `https://opencv.org/`

id	segm_start	segm_end	pp	sp	mc	img_name
S13E01_0	00:00:49	00:02:18	0	6	0	S13E01_1.jpg
Meredith: Don't you wish you could just take it back... That thing you said, that thing you did. [...] We can't undo the past. 'Cause the future keeps coming at us.						
id	segm_start	segm_end	pp	sp	mc	img_name
S13E01_1	00:02:18	00:02:36	0	2	4	S13E01_1.jpg
[Siren wails] Isaac: What do we got? We got a male, mid 20s. [...] We'll need a CT. All right, let's get him to Trauma One. Let's go. Page Avery!						
id	segm_start	segm_end	pp	sp	mc	img_name
S13E01_2	00:02:36	00:03:18	0	6	0	S13E01_2.jpg
Two champagnes. You got it. I thought you were dancing with Maggie. [...] Take a breath. What happened to DeLuca?						

Table 3.3: Some instances from the resulting corpus. The text obtained from the subtitles has been shortened for displaying purposes.

Figure 3.1: Keyframe S13E01_0.jpg.

3.2.3 Data Preprocessing and Description

The preprocessing of the corpus involved several steps designed to refine and improve the quality of the data. Most importantly, short segments (≤ 9 subtitles), in which stopwords[4] and consecutive repeated words constituted more than 65% of the total tokens, were removed. In addition to this, other preprocessing steps included:

- Removing song lyrics, e.g., ♪ *I don't want to wait...*♪
- Removing song names, e.g., *[Lorde's "Team" playing]*
- Removing subtitle authors' names, e.g., *Telescript by Raceman*, *Subtitles/Sync by Bemused.*
- Removing italics tags (`<i></i>`) without removing the content inside them, e.g., *I'm* `<i>really</i>` *sorry.*
- Removing hesitations in the characters' speech, e.g., *He-he doesn't... He doesn't mean that.*
- Removing hyphens indicating dialogue between different characters, e.g., *-Is he talking? -Yeah.*
- Removing segments containing only sounds, e.g., *[Whistles].*

Labels were also preprocessed as part of the data preparation, with the original range of [0, 6] discretized into binary values of {0, 1}. Values in the interval [0, 2] were assigned to 0 and values in the interval [4, 6] were assigned to 1; as a result, segments with label combinations 330, 303, and 033[5] were removed as they could not be discretized into the required binary representation. The counts of the instances per class before and after discretization are illustrated in Table 3.4. Although some of the granularity in the original

[4]From the NLTK library `https://www.nltk.org/`.

[5]In other words, where PP=3, SP=3, MC=0 and so on.

Before Discretization				After Discretization			
Values	PP	SP	MC	Values	PP	SP	MC
0	13668	8718	11641				
1	321	310	245	0	13690	8907	11381
2	667	751	621				
4	368	445	394				
5	156	297	233	1	3299	8082	5608
6	2775	7340	4981				

Table 3.4: Label counts before and after discretization.

No. of subtitles	1-9	10-18	19-27	28-37	38-46	47-55	56-74
No. of segments	5928	5527	3245	1572	482	176	59

Table 3.5: Distribution of subtitles per segment.

data is lost, the main advantage of this approach is that it simplifies the classification task by reducing the number of classes, which enables the model to concentrate on identifying those segments where there is a complete or mostly complete correspondence to one of the isotopies.

The final corpus used in this study contains 276,357 subtitles, which are grouped into 16,989 labeled segments. The corpus has a total of 2,260,655 tokens (38,629 types) and the mean length of a subtitle is 8.430 ± 3.921 tokens. Each segment consists of 1 to 74 subtitles, and about 95.7% of the segments (16,272) contains up to 37 subtitles, as shown in Table 3.5.

3.3 Evaluation Metrics

Classification metrics play a critical role in evaluating the performance of a classification model, i.e., how effectively it can predict the correct class or category for new, unseen data. In this section, we provide an overview of some commonly used classification metrics, explaining not only what these metrics represent, but also why they were chosen or not.

3.3.1 Accuracy

Accuracy is a commonly used metric to evaluate classification models. It is defined as the ratio of the correctly classified instances to the total number of instances. In mathematical terms, accuracy is given by:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{3.1}$$

However, accuracy can be misleading when dealing with imbalanced datasets, as a high accuracy may be obtained by simply predicting the majority class.

3.3.2 Precision and Recall

Precision and recall are two metrics that address the limitations of accuracy, especially for imbalanced datasets. Both precision and recall are calculated on the basis of a confusion matrix as the one illustrated in Figure 3.2.

		Predicted		
		Class 1	Class 2	Class 3
True	Class 1	8	2	0
	Class 2	1	6	3
	Class 3	0	1	9

Figure 3.2: Example of a confusion matrix.

In Figure 3.2, each row depicts the actual classes, and each column shows the predicted classes. The values on the main diagonal (from the top-left to the bottom-right) represent true positives (TP), where the model's predictions coincide with the actual classes. For example, 8 instances were accurately classified as Class 1 (TP for Class 1).

The elements outside the main diagonal in the matrix represent incorrect predictions. They include false positives (FP) and false negatives (FN). For instance, the value of 2 in the first row and second column means that two

instances of Class 1 were mistakenly classified as Class 2, which makes them false negatives for Class 1 and false positives for Class 2.

Precision for a class is calculated as the ratio of true positives (TP) for that class to the sum of true positives and false positives (FP) for that class. According to the confusion matrix in Figure 3.2, there are 8 true positive instances for Class 1 and 1 false positive instance where Class 2 was inaccurately predicted as Class 1. Hence, precision for Class 1 is:

$$\text{Precision}_1 = \frac{TP}{TP + FP} = \frac{8}{8+1} = \frac{8}{9} \approx 0.89 \tag{3.2}$$

On the other hand, recall is the ratio of true positives (TP) to the sum of true positives and false negatives (FN) for a particular class. As per the above confusion matrix, there are 8 true positive instances for Class 1, 2 false negatives where Class 1 was incorrectly predicted as Class 2, and 0 false negatives where Class 1 was misclassified as Class 3. Hence, recall for Class 1 is:

$$\text{Recall}_1 = \frac{TP}{TP + FN} = \frac{8}{8+2} = \frac{8}{10} = 0.8 \tag{3.3}$$

3.3.3 F_1 Score

The F_1 score is the harmonic mean of precision and recall. It balances both metrics, providing a single value that considers the trade-offs between them. To compute the F_1 score for multiclass classification, one must first calculate the precision and recall for each class. Once the precision and recall for each class have been calculated, the F_1 score can be computed using the following formula:

$$\text{F}_{1i} = 2 \cdot \frac{\text{Precision}i \cdot \text{Recall}i}{\text{Precision}i + \text{Recall}_i} \tag{3.4}$$

where F_{1i} denotes the F_1 score for class i, and $precision_i$ and $recall_i$ represent the precision and recall for class i, respectively.

For multiclass classification tasks, the F_1 scores for each class are then averaged to obtain a single, overall performance metric. There are several

ways to average the F_1 score for multiclass classification, one of which is macro-averaging, i.e. the average of the F_1 scores of each class:

$$F_{1macro} = \frac{1}{N} \sum_{i=1}^{N} F_{1_i} \tag{3.5}$$

where F_{1macro} is the macro-averaged F_1, N is the number of classes, and F_{1i} is the F_1 score for the i-th class.

In our experiments, macro-averaged F_1 will be used to measure the overall performance of the models. The main reason behind this choice is that macro-averaged F_1 gives equal weight to each class, regardless of the number of instances. This can be particularly useful when dealing with imbalanced datasets, as it ensures that the model performs well across all classes, and not just the majority class.

Chapter 4

Experiments

To address our research questions, we explore two different classification approaches to determine which one is better suited for the problem at hand: the first approach is to employ a single multiclass classifier that directly predicts the target class labels, while the second involves using the one-vs-the-rest approach to decompose the multiclass task using a series of binary classifiers. Although, as we will discuss in Section 4.1, the one-vs-the-rest approach is expected to achieve better separation between the classes, the multiclass approach would probably require less training time.

We investigate the multiclass and one-vs-the-rest approaches for both unimodal and multimodal settings. For the multiclass approach, we first fine-tune and evaluate a unimodal textual and a unimodal visual model, and then a multimodal one. For the one-vs-the-rest approach, we do the same for each unimodal binary sub-problem, and then repeat the problem decomposition approach in the multimodal setting as well. An overview of the models is presented in Section 4.2. Lastly, we report on the obtained results and proceed to answer the research questions in Section 4.3.

4.1 Classification Approaches

In this section, we examine two approaches to multiclass classification: the direct multiclass approach and the one-vs-the-rest approach. Although both methods address the challenge of categorizing instances into multiple classes, their performance can vary depending on the nature of the data and the choice of the base classifier (Al-Essa and Appice, 2021; Vera et al., 2021).

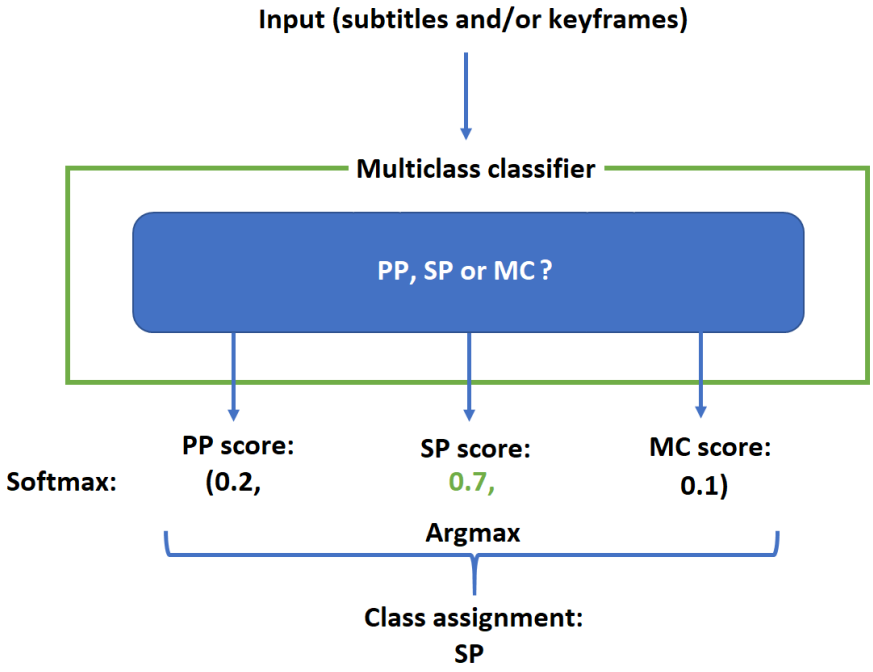


Figure 4.1: Multiclass approach; adapted from `https://cran.r-project.org/web/packages/multiclassPairs/vignettes/Tutorial.html`

4.1.1 Direct Multiclass Approach

Some algorithms, including neural networks, are capable of addressing multiclass problems directly. In a multiclass classification setting, the last layer of a neural network is usually set to be a softmax function so that the output is a probability distribution over the N output classes, as in Figure 4.1 (cfr. Section 2.1). The argmax of these scores, i.e. the class with the highest probabilty, is the final predicted class.

This approach has several advantages, including greater computational efficiency and the ability to capture relationships between the classes. However, it can also lead to worse predictive performance when dealing with imbalanced datasets and complex class boundaries, potentially causing the model to focus on the dominant classes (Ghosh et al., 2021).

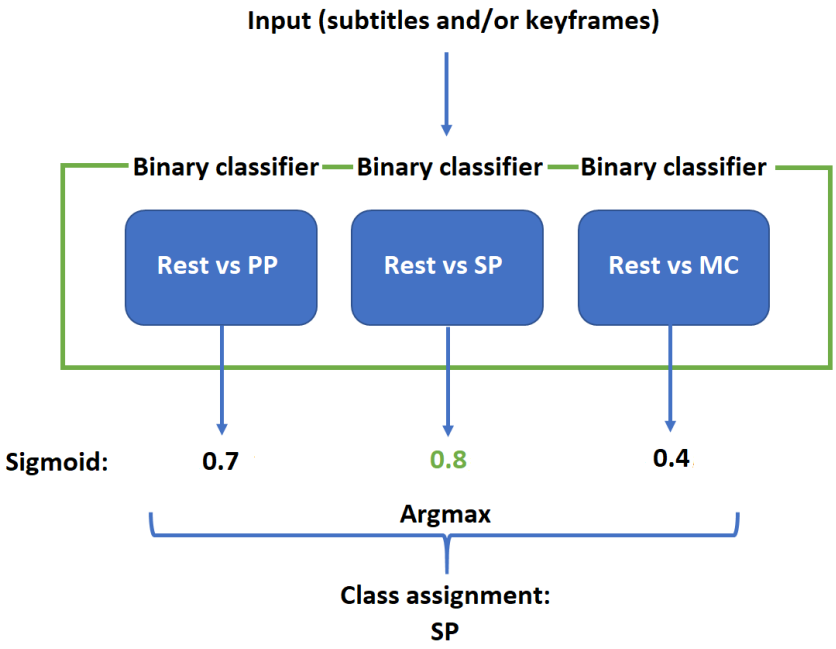


Figure 4.2: One-vs-the-rest; adapted from `https://cran.r-project.org/web/packages/multiclassPairs/vignettes/Tutorial.html`

4.1.2 One-vs-the-Rest Approach

Another way to tackle a multiclass problem is to use the one-vs-the-rest (OvR) approach, also known as one-vs-all (OvA), which involves decomposing the task into N binary classifiers, each trained to distinguish between one class and the rest (Lorena et al., 2008). As shown in Figure 4.2, the activation function used in the output layer of each binary classifier is typically a sigmoid (cfr. Section 2.1). In this setup, each binary classifier outputs a probability which represents the likelihood that the instance belongs to its associated positive class instead of all other classes. In other words, the input is evaluated by all N classifiers (where N is the number of classes), each assigning a probability that the instance belongs to its respective class. The final prediction is made by selecting the class associated with the classifier that outputs the highest probability (Aly, 2005).

Although neural networks can handle multiclass classification directly, using the one-vs-the-rest strategy may be beneficial in certain situations. In

some cases, multiclass classification may lead to a higher rate of classification errors due to its increased complexity compared to binary classification (Lorena et al., 2008). The one-vs-the-rest approach can reduce this complexity, potentially resulting in a more discriminative model (Vogiatzis et al., 2023). This approach also presents benefits in that it allows the combination of different models for each class; however, it is also more computationally expensive, as it requires N models to be trained (Sánchez-Maroño et al., 2010).

4.2 Models

In this section, we present the models that are used to address the research questions at the core of this thesis. We begin by providing an overview of BERT (Devlin et al., 2019), a pre-trained transformer model that has revolutionized the field of NLP by setting new benchmarks for a wide range of tasks. Following the discussion on BERT, we shift our focus to CLIP, a recent model for visual feature extraction and understanding (Radford et al., 2021), and MMBT (Kiela et al., 2019), a multimodal extension of BERT that integrates visual and textual information.

4.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a highly performing pre-trained language model introduced by Devlin et al. (2019). Built on the transformer architecture, BERT introduces a bidirectional pre-training approach, which sets it apart from unidirectional models like GPT (cfr. Section 2.1.2). Using a WordPiece tokenizer[1], BERT is pretrained on two tasks — masked language modeling (MLM) and next sentence prediction (NSP) — to learn rich contextual information and sentence relationships (cfr. Section 2.1.2), leveraging a large corpus of text that includes BooksCorpus, a dataset of over 11,000 books, and English Wikipedia with around 2.5 billion words.

Figure 4.3 shows BERT in a text classification setting. The input sequence consists of tokens (Tok 1, Tok 2, ... Tok N) and a special [CLS] token. These tokens are first converted into static embeddings ($E_{[CLS]}$, E_1,

[1] `https://github.com/google-research/bert#tokenization`

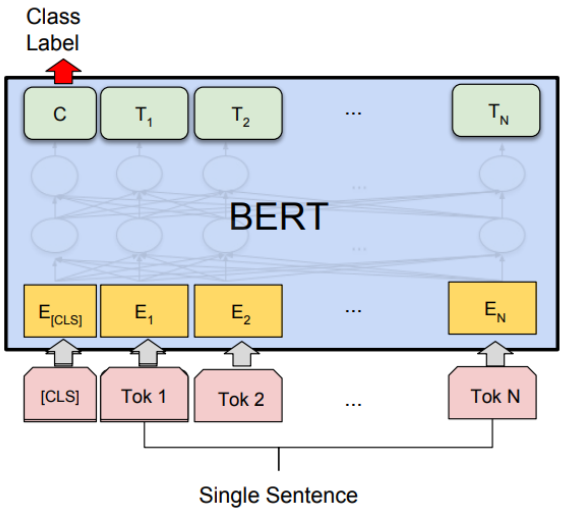


Figure 4.3: BERT for text classification (Devlin et al., 2019).

E_2 ... E_N). At this stage, the embeddings do not capture any contextual information from the surrounding tokens in the input sequence. Then, they are contextualized within BERT. After processing, the contextualized embeddings of the tokens (T_1, T_2, T_3, ... T_N) and the [CLS] token, now denoted as 'C', are produced. The contextualized token 'C' captures the global information of the input sequence and is used for tasks requiring a single, aggregated representation. In the case of classification, a task-specific classification head is typically added on top of BERT to map the contextualized representations to class probabilities (Devlin et al., 2019).

For the unimodal experiments, we use the `bert-base-uncased` model from the HuggingFace library[2]. The model is fine-tuned (cfr. Section 2.1.2) exploring epochs $\in [1, 2, 3]$ with a batch size of 16, one of the batch sizes recommended by the authors of BERT (Devlin et al., 2019). For optimization, we employ the AdamW[3] optimizer with a learning rate of 1e-5 and an epsilon value of 1e-8. We encode the training, validation, and test datasets with BertTokenizer[4] and pad the sequences to a maximum length of 512.

[2] `https://huggingface.co/bert-base-uncased`

[3] `https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html`

[4] `https://huggingface.co/transformers/v3.0.2/model_doc/bert.html#berttokenizer`

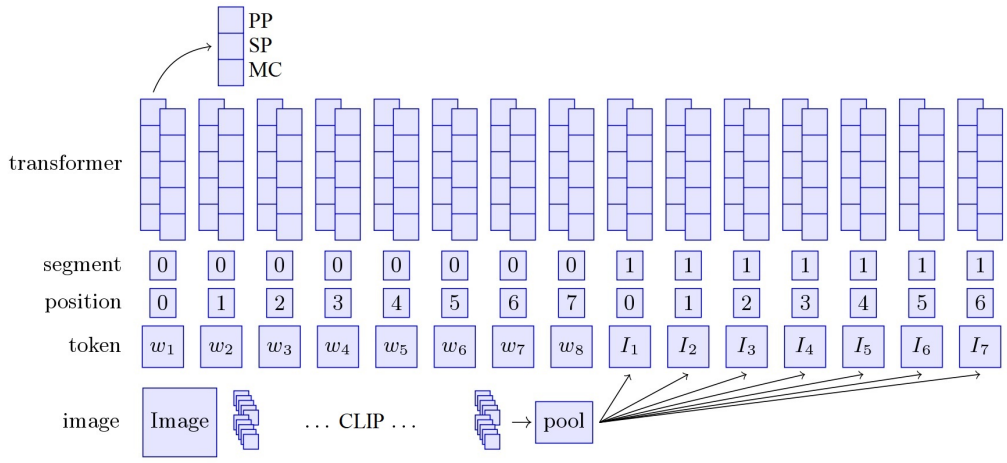


Figure 4.4: Representation of the multimodal bitransformer architecture combining CLIP and BERT; adapted from Kiela et al. (2019).

To adapt the unimodal model for the approaches outlined in Section 4.1, we modify the `num_labels` parameter of BERT, setting it to 3 for multiclass classification and 1 for binary classification. For multiclass, we use the default Cross Entropy loss function that is computed by BERT when `num_labels` $>$ `1` (Devlin et al., 2019). For binary classification in the one-vs-the-rest scenario, we use the Binary Cross Entropy with Logits loss from PyTorch[5].

4.2.2 CLIP and MMBT

The Multimodal Bitransformer (MMBT) is an architecture that extends the capabilities of bidirectional transformers, like BERT, to handle multimodal data. Introduced by Kiela et al. (2019), MMBT incorporates the strengths of the transformer architecture and adapts it for processing both textual and visual inputs, enabling the model to effectively learn from and generate meaningful outputs for multimodal scenarios. The idea behind MMBT is to jointly fine-tune an unimodally pretrained text encoder, such as BERT, and an image encoder, typically based on a CNN, such as ResNet (cfr. Section 2.2). As illustrated in Figure 4.4, the MMBT model achieves this by

[5] `https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss`

projecting the image embeddings into the text token space. Here, textual segments are marked with a 0, while visual segments are indicated with 1. By processing both segments at the same time, attention mechanisms (cfr. Section 2.1.2) can effectively operate across both modalities. Tokens are indexed based on their positions, ranging from 0 up to a maximum of 512, while image representations are indexed from 0 to 640.

To further enhance the capabilities of MMBT, we use OpenAI's CLIP (Radford et al., 2021) as the visual encoder instead of the default ResNet-152 architecture used by MMBT. Following Neskorozhenyi (2021), we use the Pillow library[6] to prepare 288x288 pixel versions of all frames by rescaling and padding, while also maintaining the original aspect ratio of the frames. We then slice the frames into three equal parts to capture both global and local image features. By following this procedure, four vectors are obtained for each frame: a vector for each of the parts that encode spatial information and one for the whole frame. The visual feature extractor of CLIP is RN50x4, a modified version of ResNet-50 which has been shown to be particularly effective for vision-and-language tasks (cf. Section 2.2).

As for the textual encoder, we again use `bert-base-uncased` so as to be able to compare the performance of MMBT and BERT. We fine-tune the MMBT architecture by exploring epochs $\in [1, 2, 3]$ with a batch size of 8 and a gradient accumulation of 20 steps to reduce memory usage. For optimization, we employ the MADGRAD[7] optimizer with a learning rate of 2e-4. As for BERT, we adhere to the preprocessing and parameters used in the unimodal textual setting. Given that MMBT is largely based on BERT's achitecture, the `num_labels` parameter and the loss functions are also configured in the same way as BERT. We maintain these choices for the unimodal visual model based on CLIP, with the exception that we do not use the textual encoder. As for the CLIP-based model, we leave RN50x4 as the feature extractor and we follow Wei et al. (2022) in using a batch size of 16.

4.3 Results

In this Section, we report the outcomes of the experiments involving CLIP, BERT, and MMBT using two different approaches: multiclass and one-vs-

[6] `https://github.com/python-pillow/Pillow`
[7] `https://github.com/facebookresearch/madgrad`

Setting	CLIP	BERT	MMBT
ovr@1	0.461 ± 0.080	$\mathbf{0.681 \pm 0.007}$	0.664 ± 0.033
multi@1	0.436 ± 0.031	0.677 ± 0.013	0.651 ± 0.017
ovr@2	0.536 ± 0.014	0.684 ± 0.010	0.686 ± 0.008
multi@2	0.487 ± 0.027	0.695 ± 0.010	$\mathbf{0.697 \pm 0.028}$
ovr@3	0.542 ± 0.072	0.686 ± 0.008	0.688 ± 0.012
multi@3	0.523 ± 0.016	0.702 ± 0.011	$\mathbf{0.719 \pm 0.011}$

Table 4.1: Macro-averaged F_1 obtained with one-vs-the-rest and multiclass approaches over 10-fold cross-validation at [1, 2, 3] epochs. Best result on epoch is in bold and best result on metric is in red.

the-rest, as described in Section 4.1 and Section 4.2. In Section 4.3.1, we report the average F_1 scores obtained from 10-fold cross-validation. Then, in Section 4.3.2, we address the key research questions at the core of this dissertation. We first examine the two approaches, namely multiclass vs one-vs-the-rest, then we determine the most informative modality for predicting the isotopies, and lastly we evaluate the effect of incorporating keyframes.

4.3.1 Evaluation

This Section provides an overview of the results obtained from each combination of models (CLIP, BERT, MMBT) and approaches (multiclass and one-vs-the-rest), as outlined in Section 4.2. In Table 4.1, we report the macro-averaged F_1 scores for each of the settings, evaluated with 10-fold cross-validation. For one-vs-the-rest, the reported score is the average F_1 obtained by the three binary models. A higher F_1 score indicates that the model is more effective at identifying true positives. The aim of the evaluation phase is to identify the best model for each configuration.

At 1 epoch, the BERT models slightly outperform the MMBT multimodal models. This early stage result can be explained by the complexity of the multimodal MMBT model, which requires more epochs to fully optimize its performance. Another interesting result is the fact that at 1 epoch, all one-vs-the-rest models outperform their multiclass counterparts. This is the only

Model	Setting	$\mathbf{F_{1_{test}}}$	$\mathbf{Recall_{test}}$	$\mathbf{Precision_{test}}$
CLIP	One-vs-rest	0.566	0.572	0.584
CLIP	Multiclass	0.536	0.545	0.545
BERT	One-vs-rest	0.685	0.684	0.687
BERT	Multiclass	0.672	0.668	0.679
MMBT	One-vs-rest	0.713	0.712	0.714
MMBT	Multiclass	**0.723**	**0.726**	**0.720**

Table 4.2: Performance on test of the best models for each configuration.

epoch where one-vs-the-rest shows better results overall. This is likely due to the fact that the individual binary models have to learn a less complex relationship between inputs and targets, resulting in higher performance after just a single epoch of training, unlike the multiclass models.

At 2 epochs, the BERT and MMBT models obtain similar results in the one-vs-the-rest configuration, with MMBT slightly outperforming BERT. The improvement of MMBT over BERT also starts to show in the multimodal multiclass setting, indicating a potential advantage in MMBT's approach to handling the complexity of the task as the model continues to learn. It is worth noting that at this stage, the one-vs-the-rest MMBT no longer outperforms its multiclass counterpart, which could be due to the increased optimization and complexity handling that the multiclass model has developed over the additional epoch of training. While CLIP obtains lower F_1 scores than both BERT and MMBT, it is also the only model that continues to show better results when using one-vs-the-rest rather than multiclass.

At 3 epochs, all models continue to show progress in terms of their performance. By the end of the third epoch, BERT and MMBT are still obtaining close results in the one-vs-the-rest configuration, but the MMBT model now clearly outperforms all others in the multiclass configuration. This suggests that as the training progresses, the MMBT model's ability to handle complex class relationships improves, thus boosting its performance. Another observation concerns BERT, where multiclass shows a more significant improvement over one-vs-the-rest than in the previous epochs. This provides additional evidence supporting the idea that more complex models, given sufficient training time, have the potential to outperform their simpler coun-

terparts. While CLIP models could have benefited from more epochs, we trained the models for three epochs as the better-performing BERT and MMBT models were starting to overfit the training data.

Following the 10-fold cross-validation, we perform a final evaluation of the best models selected on the basis of the cross-validation on an independent test set that was not utilized during training. This final evaluation allows us to assess the models' performance on unseen data and their generalization capabilities. F_1 score, Precision and Recall of the best configurations on the test set are reported in Table 4.2. From the test results, it emerged that ensembling the best-performing binary models in the one-vs-the-rest scenario resulted in this approach outperforming multiclass also in the unimodal textual scenario. As for the remaining configurations, the trends identified during cross-validation are confirmed in the final evaluation. All of the models that were selected for the final evaluation obtained their highest F_1 at epoch three.

4.3.2 Discussion

Building upon the findings presented in Section 4.3.1, we now shift our attention towards addressing the research questions formulated in Section 3.1.

As for RQ1, the answer is not straightforward. The approach which resulted in the best-performing model is the direct multiclass approach. Specifically, multiclass with MMBT and trained over 3 epochs achieved the highest average F_1 score on the test set, i.e. $F_1 = 0.723$. This result could be attributed to the ability of the multiclass MMBT approach to better handle correlations between different classes, a feature not captured by the one-vs-the-rest approach which treats each class independently (cf. Section 4.1). It is possible that the added visual information allows MMBT to disambiguate instances more effectively than the multiclass the BERT model, resulting in one-vs-the-rest being more effective for BERT (F_1 of 0.684 for OvR compared to 0.668 for multiclass). What emerged during the evaluation phase (cf. Table 4.1) is that one-vs-the-rest can provide an early advantage due to the decomposition of the multiclass problem into multiple binary problems; however, as the learning process continues, both approaches can be effective depending on the model. In this sense, the evaluation on the test set proved to be crucial in order to understand the potential of one-vs-the-rest when ensembling the individual binary models.

Class	Setting	F_1 CLIP	F_1 BERT	F_1 MMBT
PP	One-vs-rest	0.444	0.563	0.579
PP	Multiclass	0.341	0.513	0.593
SP	One-vs-rest	0.708	0.788	0.824
SP	Multiclass	0.724	0.782	0.809
MC	One-vs-rest	0.592	0.706	0.741
MC	Multiclass	0.604	0.719	0.765

Table 4.3: Per-class one-vs-the-rest and multiclass F_1 scores on test.

Interestingly, CLIP also benefits from the one-vs-the-rest approach. Upon closer inspection, it appears that the reason why multiclass CLIP has a lower F_1 score is that it underperforms on the minority class, i.e. the professional plot (PP). As illustrated in Table 4.3, this is in contrast with one-vs-the-rest, which obtains a higher overall F_1 score. Even though one-vs-the-rest obtains a lower F_1 score than multiclass on the sentimental plot (SP) and medical case (MC) classes, it manages to identify PP instances significantly better (one-vs-the-rest achieves an F_1 score of 0.444 on PP compared to the 0.341 of multiclass). This could be due to the fact that one-vs-the-rest is more suitable for unimodal models, as a similar trend also arises when it comes to multiclass BERT compared to one-vs-the-rest BERT. Given that both approaches can perform well overall, we will proceed to answer RQ2 and RQ3 by analyzing the best performing approach for each modality, i.e. one-vs-the-rest CLIP and BERT and multiclass MMBT.

In order to address RQ2, we will now compare the two unimodal models to determine which modality is more informative for the task of predicting the isotopies. On the test set, one-vs-the-rest BERT achieves an F_1 score of 0.685, while one-vs-the-rest CLIP obtains a fairly lower F_1 score of 0.566 (cf. Table 4.2). As for RQ2, we can conclude that BERT performs better than CLIP, which suggests that the text might be more informative than the keyframes for the task of predicting the isotopies. It should be noted, however, that the CLIP model is limited by the fact that it takes into consideration a single keyframe for each segment. Considering the average length of the texts available to BERT (cf. Section 3.2.3), it is clear that the textual model not only has access to more information but can also

analyze dialogue at different points in time, unlike the CLIP model which looks exclusively at the midframe of a segment. Considering the importance of textual data for predicting the isotopies, it could be promising to explore multilingual transformer-based models like mBERT (Devlin et al., 2019) or XLM-RoBERTa (Conneau et al., 2020). Given the availability of subtitles in other languages, this would allow an investigation into how these models generalize in multilingual scenarios.

Moving on to RQ3, we proceed to assess whether the combination of keyframes and subtitles results in higher performance by examining the results of MMBT compared to BERT. As shown in Table 4.2, the best-performing MMBT model, i.e. multiclass MMBT, obtains an F_1 score of 0.723, compared to one-vs-the-rest BERT's F_1 score of 0.685. It should be noted that multiclass MMBT's F_1 score of 0.723 is the highest across all models and configurations. Considering RQ3, we can conclude that using a multimodal approach results in a noticeable improvement over the text-only BERT model. As already mentioned, it must be taken into account that the frames were selected somewhat arbitrarily, with only one frame taken from each segment. We can therefore consider this result to be promising, as it demonstrates that integrating more information from the visual channel can improve the performance of the model. To overcome this limitation, an approach that takes into consideration multiple frames or a more systematically chosen single frame could be developed. Another avenue of research could involve exploring the type of vision-and-language model that is used: MMBT is a single-stream architecture (Kiela et al., 2019); however, existing research suggests that dual-stream models can obtain better results thanks to their co-attention mechanism (cf. Section 2.3). As dialogue is typically not a descriptive account of the ongoing events, but rather an interaction between characters, the correlation between the modalities might be too complex for an early-fusion based model.

In summary, the results suggest promising directions for future work in the application of unimodal and multimodal models for the automatic identification of isotopies. One-vs-the-rest appears to be more effective for unimodal models, while textual features proved to be more informative than keyframes for predicting the isotopies as BERT outperformed CLIP with an F_1 score of 0.685 as compared to CLIP's 0.566. Finally, the improvement obtained by the multimodal MMBT model over the text-only BERT model provides support for the initial hypothesis that the information from the vi-

sual channel complements the one that is contained in the dialogues. These results also indicate the potential of single-frame approaches for the task of multimodal video classification. However, considering the strategies adopted by models such as Singularity and ATP (cf. Section 2.4), it could be possible to obtain even better results by devising a more systematic methodology for frame selection instead of relying exclusively on the midframes. Additional avenues for further research might include the adoption of a dual-stream model as an alternative to the single-stream architecture of MMBT, or the inclusion of more frames for each segment. In the broader context, we can conclude that automated content analysis for isotopy identification, a domain which has been previously unexplored, can greatly benefit from multimodal approaches. Exploring the use of multilingual transformer-based models like mBERT (Devlin et al., 2019) or XLM-RoBERTa (Conneau et al., 2020) could also lead to improvements and open up the possibility of leveraging cross-lingual transfer for the analysis of other shows pertaining to the medical drama genre.

Chapter 5

Conclusions

This dissertation examined three research questions to evaluate various methods for automatic isotopy identification in the context of medical dramas. The first research question focused on comparing, for all models, the performance of a direct multiclass approach versus a one-vs-the-rest approach. The second research question aimed to determine the most informative modality for the classification task. The third research question involved investigating whether the inclusion of keyframes in addition to subtitles resulted in better performance compared to just using the subtitles. The motivation behind these research questions is related to the growing interest in using computational methods for analyzing complex audiovisual contents, including long-running medical dramas such as Grey's Anatomy. Motivated by the hypothesis that a multimodal model incorporating both textual and visual data would outperform a unimodal model trained solely on text, we created a multimodal corpus by expanding on the Medical Dramas Dataset, which includes segments annotated with the corresponding isotopies from eight TV shows pertaining to the medical drama genre. Textual features were extracted by temporally aligning the subtitles with the segments, while visual features were obtained by extracting a frame, referred to as a keyframe, between the start and the end of each segment. The obtained multimodal corpus comprises 17 seasons from the show Grey's Anatomy, for a total of 6,989 labeled segments and 2,260,655 tokens. We then used this corpus to experiment with both unimodal and multimodal transformer-based models, namely CLIP, BERT, and MMBT, aiming to understand how different modalities and approaches to the classification task can impact the identification of the isotopies.

The findings from this work are promising, indicating that it is indeed possible to utilize deep learning to automatically identify the distinctive isotopies of the medical drama genre in the context of Grey's Anatomy. Notably, we observed that the multimodal MMBT model performed significantly better compared to the text-only BERT model and the image-only CLIP model. More specifically, MMBT achieved the top F_1 score of 0.723, compared to BERT's highest F_1 score of 0.685, thus shedding light on the potential benefit of incorporating visual information alongside textual data. We have also examined different approaches to the problem, observing that the one-vs-the-rest approach appears to be more beneficial in the case of unimodal models. It is possible that the added visual information allows MMBT to disambiguate instances more effectively than the multiclass the BERT model, which could explain why this is the only setting in which multiclass worked better than one-vs-rest. Lastly, we also concluded that the textual information proved to be more informative than the visual data, demonstrating the importance of dialogue for isotopy identification. Hence, when computational resources are a limiting factor, a text-based model can also be a valid approach.

The potential for future work is vast, as there are many aspects that could be further improved to enhance the performance of the models. For example, future research could delve into a more systematic methodology for frame selection, which in this study was limited to only the midframes of the segments. A comprehensive approach that takes into consideration multiple frames or systematically chosen single frames could potentially lead to significant improvements in the performance of the models. Additionally, while the single-stream architecture of MMBT model provided promising results, exploring dual-stream models might result in even better performance. Existing research suggests that dual-stream models can obtain better results thanks to their co-attention mechanism, which enables them to handle complex relationships between the modalities. Moreover, the potential to leverage cross-lingual transfer for the analysis of other shows pertaining to the medical drama genre could be explored. This could be achieved by experimenting with multilingual transformer-based models like mBERT or XLM-RoBERTa. Given the availability of subtitles in other languages, this approach would allow for an investigation into how these models generalize in multilingual scenarios.

Bibliography

Malik Al-Essa and Annalisa Appice. Dealing with imbalanced data in multiclass network intrusion detection systems using xgboost. *Communications in Computer and Information Science*, pages 5–21, 2021.

Mohamed Aly. Survey on multiclass classification methods. *Technical Report*, 19:1–9, 2005.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, pages 423–443, 2018.

Khaled Bayoudh, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa. A survey on deep multimodal learning for computer vision: Advances, trends, applications, and datasets. *The Visual Computer*, pages 1–32, 2022.

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the "video" in video-language understanding. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2917–2927, 2022.

Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language berts. *Transactions of the Association for Computational Linguistics*, 9:978–994, 2021.

François Chollet. *Deep Learning with Python*. Manning Publications, 2020.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

Jacob Devlin, Chang Ming-Wei, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. A survey of vision-language pre-trained models. *arXiv preprint arXiv:2202.10936*, 2022.

Mohamed Elgendy. *Deep Learning for Vision Systems*. Manning Publications, 2020.

Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. Semeval-2022 task 5: Multimedia automatic misogyny identification. *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 533–549, 2022.

Kushankur Ghosh, Colin Bellinger, Roberto Corizzo, Bartosz Krawczyk, and Nathalie Japkowicz. On the combined effect of class imbalance and concept complexity in deep learning. *2021 IEEE International Conference on Big Data*, pages 4859–4868, 2021.

Wenzhong Guo, Jianwen Wang, Wang, and Shiping. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.

Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27,48, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Elike Hodo, Xavier Bellekens, Ephraim Iorkyase, and Andrew Hamilton. Machine learning approach for detection of nontor traffic. *Proceedings of*

the 12th International Conference on Availability, Reliability and Security, pages 1, 6, 2017.

Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. What makes multi-modal learning better than single (provably). *Advances in Neural Information Processing Systems*, 34:10944–10956, 2021.

Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950*, 2019.

Klaus Krippendorff. *Content analysis: an introduction to its methodology*. SAGE, 1995.

Hobson Lane, Cole Howard, and Hannes Hapke. *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*. Manning Publications, 2019.

Jie Lei, Tamara Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022.

Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, and Yu Cheng. Value: A multi-task benchmark for video-and-language understanding evaluation. *arXiv preprint arXiv:2106.04632*, 2021.

Liunian Li, Harold Mark Yatskar, Da Yin Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

Chang Liu, Armin Shmilovici, and Mark Last. Mnd: A new dataset and benchmark of movie scenes classified by their narrative function. *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 610–626, 2023.

Jingzhou Liu, Wenhu Chen, Yu Cheng, Zhe Gan, Licheng Yu, Yiming Yang, and Jingjing Liu. Violin: A large-scale dataset for video-and-language inference. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10900–10910, 2020.

Ana Carolina Lorena, André De Carvalho, and João Gama. A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, 30:19–37, 2008.

Jiansen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

Arianna Muti, Katerina Korre, and Alberto Barrón-Cedeño. Unibo at semeval-2022 task 5: A multimodal bi-transformer approach to the binary and fine-grained identification of misogyny in memes. *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 663–672, 2022.

Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10317–10326, 2020.

Rostyslav Neskorozhenyi. How to get high score using mmbt and clip in hateful memes competition. *Towards Data Science*, 2021.

Guglielmo Pescatore and Marta Rocchi. Narration in medical dramas: Interpretative hypotheses and research perspectives. *La Valle dell'Eden*, 1: 107–115, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Kruegen, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *International conference on machine learning*, pages 8747–8763, 2021.

Vijaya Teja Rayavarapu, Bharath Bhat Myra Nam, Vikas Bahirwani, and Shobha Diwakar. Multimodal transformers for detecting bad quality ads on youtube. *Proceedings of The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2022 (AdKDD '22)*, 2019.

Marta Rocchi and Guglielmo Pescatore. Modeling narrative features in tv series: coding and clustering analysis. *Humanities and Social Sciences Communications*, 9(333), 2022.

Soo Hyun Ryu and Richard Lewis. Accounting for agreement phenomena in sentence comprehension with transformer language models: Effects of similarity-based interference on surprisal and attention. *arXiv preprint arXiv:2104.12874*, 2021.

Sheng Shen, Liunian Harold Li, Hao Tan, Bansal Mohit, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7464–7473, 2019.

Noelia Sánchez-Maroño, Amparo Alonso-Betanzos, Pablo García-González, and Verónica Bolón-Canedo. Multiclass classifiers vs multiple binary classifiers using filters for feature selection. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2010.

Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

Makarand Tapaswi. Story understanding through semantic analysis and automatic alignment of text and video. *Doctoral dissertation, Karlsruhe Institute of Technology*, 2016.

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Aligning plot synopses to videos for story-based retrieval. *International Journal of Multimedia Information Retrieval*, 4(1), 2014.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. O'Reilly Media, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jacob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Daniel Vera, Oscar Araque, and Carlos Iglesias. *GSI-UPM at IberLEF2021: Emotion Analysis of Spanish Tweets by Fine-tuning the XLM-RoBERTa Language Model*. 2021.

Antonios Vogiatzis, Stavros Orfanoudakis, Georgios Chalkiadakis, Konstantia Moirogiorgou, and Michalis Zervakis. Novel meta-learning techniques for the multiclass image classification problem. *Sensors*, 23(1), 2023.

Lanxiao Wang, Wenzhe Hu, Heqian Qiu, Chao Shang, Taijin Zhao, and Benliu Qiu. A survey of vision and language related multi-modal task. *CAAI Artificial Intelligence Research*, 1(2):111–136, 2022.

Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. *arXiv preprint arXiv:2205.14141*, 2022.

Zejia Weng, Lingchen Meng, Rui Wang, Zuxuan Wu, and Yu-Gang Jiang. A multimodal framework for video ads understanding. *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4843–4847, 2021.

Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry Davis. Adaframe: Adaptive frame selection for fast video recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

Peng Xu, Xiatian Zhu, and David Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Sayyed Zahiri and Choi Jinho. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. *arXiv preprint arXiv:1708.04299*, 2017.

Lisai Zhang, Hongfa Wu, Qingcai Chen, Yimeng Deng, Joanna Siebert, Zhonghua Li, Yunpeng Han, Dejiang Kong, and Zhao Cao. Vldeformer:

Vision–language decomposed transformer for fast cross-modal retrieval. *Knowledge-Based Systems*, 252, 2022.

Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755, 2020.