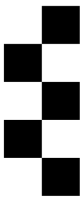
POLYTECHNIC UNIVERSITY OF TURIN

UNIVERSITY OF ESSEX

Politecnico di Torino



University of Essex

Master's Degree in Data Science and Engineering

Master's Degree Thesis

Decoding Dreams and Thoughts using AI Sentiment Analysis

Supervisors

Prof. Vito DE FEO

Prof. Valeria CHIADO' PIAT

Candidate

Mahdi HASANZADEH

October 2025

Summary

Understanding and classifying the emotional content of dream descriptions is a unique challenge that sits between psychology and natural language processing. This thesis tackles this issue by creating a machine learning pipeline that can automatically predict sentiment labels from free-text dream narratives.
The main method uses handcrafted lexical-semantic features for sentiment classification. Specifically, Term Frequency–Inverse Document Frequency (TF-IDF) extracts lexical patterns, while Empath, a psychologically grounded lexicon, provides high-level semantic signals. To address extreme class imbalance in the labeled data, a targeted resampling strategy based on SMOTE is used to ensure balanced representation across all sentiment categories.
A Multi-Layer Perceptron (MLP) classifier is trained on the engineered features and evaluated using both 5-fold stratified cross-validation and Leave-One-Out Cross-Validation (LOOCV). The model achieves a high accuracy of 86% and strong macro F1-scores across folds and samples. This shows robust generalization with minimal overfitting. In addition to this main pipeline, the thesis also looks into fine-tuning large language models. A DeBERTa transformer was adapted using parameter-efficient LoRA (Low-Rank Adaptation), allowing for modeling emotional language in context. Although this was not integrated into the main system, this parallel investigation highlights the potential of transformer-based models for understanding emotions in narrative text. Beyond model performance, the thesis examines how consistent the sentiment labels are by comparing multiple sources. Cross-dataset accuracy matrices and correlation analyses (Cohen's Kappa and Spearman's rho) show significant disagreements among annotators. This highlights the subjective nature of emotional labeling in dreams.
Overall, this work provides a reproducible affective computing pipeline for narrative data and offers insights into the challenges of subjectivity in human-labeled sentiment datasets, especially in areas like dream interpretation.

Acknowledgements

I would like to express my deepest gratitude to Prof. Vito De Feo, whose guidance, patience, and trust made this work possible. His insight and support have been invaluable not only for this research but also for my personal and academic growth.

My heartfelt thanks also go to Prof. Valeria Chiado' Piat, for her continuous encouragement, thoughtful feedback, and inspiring perspective throughout this journey. Her kindness and clarity helped me navigate moments of doubt with confidence.

I am sincerely thankful to Dr. Poerio for her precious help and collaboration in the dataset development phase. Her expertise and willingness to assist were fundamental to the progress of this project.

Finally, I would like to thank my family and friends, for their endless love, patience, and belief in me. Their presence, even from afar, has been my greatest source of strength and comfort during every step of this journey.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Understanding the emotional content of human experiences has long been a topic of interest in both psychology and computational fields. Among these experiences, dreams present a particularly rich yet underexplored form of subjective expression. Free-text dream descriptions offer insights into the subconscious but pose significant challenges for computational modeling due to their unstructured, symbolic, and emotionally nuanced nature. This thesis focuses on the application of natural language processing (NLP) and machine learning techniques to automatically decode the sentiment embedded in such dream narratives.

The central goal of this research is to develop a sentiment analysis framework capable of classifying dream descriptions into discrete emotional categories. This problem is made complex by several factors, including subjective annotation variability, semantic ambiguity in dream language, and severe class imbalance in available labeled datasets. Moreover, existing models in affective computing often overlook the specific challenges posed by dream data, such as its abstract vocabulary and lack of contextual anchors.

To address these challenges, this work proposes a hybrid feature extraction pipeline that combines TF-IDF (to capture lexical frequency patterns) and Empath (to represent high-level semantic categories). To mitigate the impact of class imbalance, a resampling strategy is applied using SMOTE. A Multi-Layer Perceptron (MLP) classifier is then trained and evaluated using both 5-fold Stratified Cross-Validation and Leave-One-Out Cross-Validation (LOOCV), achieving consistently high performance across multiple metrics.

Beyond the technical implementation, this thesis also examines the quality and consistency of sentiment annotations across multiple sources, such as participant self-ratings, external judges, and research-generated labels. The study includes a correlation and cross-dataset generalization analysis to explore whether models trained on one annotation source can generalize to others. The findings reveal major discrepancies in labeling standards, which significantly affect model transferability

and highlight the subjectivity inherent in emotion labeling tasks.

This work contributes not only a high-performing sentiment classification model but also a deeper understanding of the limitations of human-annotated affective data. By focusing on dream narratives, a domain deeply tied to emotion and meaning. This research aims to support future efforts in psychological AI, personalized mental health technologies, and the broader study of human internal states through language.

Chapter 2

Literature Review

Dreams, especially those that occur during REM (rapid eye movement) sleep, have long been viewed as emotionally rich and personally meaningful experiences. According to Hartmann [1], dreaming may help us process emotions by symbolically connecting our feelings and memories. Rather than showing events exactly as they happened, dreams tend to highlight what is emotionally significant, even if the dream content appears strange or disconnected.

Neuroscience research supports this view. Studies show that during REM sleep, brain regions involved in emotion, such as the limbic system, become more active, while areas responsible for logic and self-control, such as the dorsolateral prefrontal cortex, are less active [2]. This pattern helps explain why dreams often feel intense, emotional, and surreal.

2.1 The Emotional Nature of Dreaming and Its Computational Analysis

More recently, researchers have started using tools from artificial intelligence and natural language processing to study dream reports. For example, Kim *et al.* [3] used computational models to analyze the emotional content of dream texts, aiming to connect narrative structure and sentiment with real psychological traits. This new approach brings together neuroscience, psychology, and AI to better understand what dreams can reveal about the human mind.

Sentiment and emotion analysis have become key tasks in natural language processing, particularly when dealing with short, subjective texts. While traditional sentiment analysis often focuses on binary or ternary classification (e.g., positive, negative, neutral), emotion classification expands this scope by identifying specific emotional states such as *joy*, *anger*, *fear*, or *sadness*. In their comparative study, *et al.* [4] evaluated various machine learning approaches for classifying emotions

from short texts such as tweets, highlighting the effectiveness of lexicon-based labeling using the NRC Emotion Lexicon in combination with classical models like SVM and logistic regression. Their results showed that term frequency-based features (e.g., TF-IDF) could outperform word embeddings in certain emotion classes, especially when the input texts are brief and context-limited. These findings are particularly relevant for analyzing dream narratives, which are often short, symbolic, and emotionally rich. By comparing multiple classifiers and feature sets, the study provides a strong methodological foundation for multi-class emotion classification tasks, supporting the choice of techniques such as TF-IDF and lexicon features in this thesis.

One of the earliest attempts to classify sentiment in dreams was proposed by Nadeau *et al.* [5], who labeled 100 dream reports using a 4-level negative sentiment scale and tested several feature extraction techniques for automated classification. They compared traditional lexicon-based tools (General Inquirer and LIWC), weighted semantic lexicons, and a Bag-of-Words (BoW) baseline. Among these, the General Inquirer achieved the highest accuracy (50%) and lowest mean squared error (0.577), while LIWC performed similarly well. In contrast, the BoW and weighted lexicons performed significantly worse. Given these results, using lexicon-based features is a **must** for our work: two of the top-performing methods relied on emotional lexicons, highlighting their strength in capturing the symbolic and emotionally rich nature of dream language. This supports our choice to incorporate tools like GI, LIWC, and Empath in our own sentiment classification pipeline.

2.2 Deep Learning in Sentiment Analysis

Deep learning has become a major force in sentiment analysis research. Tang, Qin, and Liu [6] review many successful deep learning methods and show that these models often outperform older approaches when there's plenty of data. However, they also emphasize that before deep learning dominated, lexicon-based features (manually built lists of sentiment words) were among the most reliable tools. Not only do lexicon methods provide strong baselines, but they also help in domains where data is scarce or where interpretability matters.

For our work on dream reports, which are symbolic, emotionally rich, and often limited in quantity, this means lexicon-based features are not optional: they are essential. Given their past top performance, we will include lexicons like Empath, alongside more modern embeddings and deep nets, to ensure both strong performance and emotional validity.

Tang, Qin, and Liu [6] provide a foundational overview in which they compare traditional methods relying heavily on feature engineering (such as sentiment

lexicons, bag-of-words, manual features) with newer deep learning approaches. They argue that while deep models (e.g. RNNs, CNNs, gated recurrent networks) have improved performance by automatically learning semantic representations from data, lexicon-based features remain highly valuable, particularly when data are limited or when interpretability is important.

Figure 2.1 illustrates the general workflow for deep learning-based sentiment analysis. The process begins with raw text, moves through tokenization and word or token embedding, and proceeds with network training via forward and backward propagation using a loss function. Transformer-based models fit neatly into this pipeline, with their embeddings and self-attention components allowing richer representation of context and meaning.

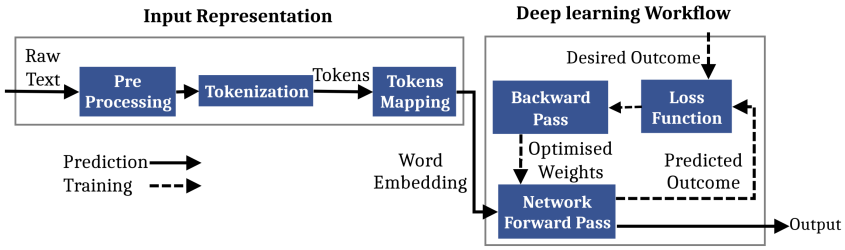


Figure 2.1: Supervised sentiment analysis workflow with deep learning. This diagram shows how raw text is preprocessed (preprocessing, tokenization), converted into embeddings, and then used in a neural network, where training includes forward and backward passes guided by a loss function.

More recently, Abdullah & Ahmet [7] show a clear shift toward transformer-based architectures such as BERT, RoBERTa, and DeBERTa in sentiment analysis. These transformer models consistently outperform earlier recurrent and convolutional networks by capturing richer contextual information, modeling long-distance dependencies, and detecting subtle emotional cues. The survey also identifies persistent challenges: domain adaptation when text styles or content shift, interpretability since deep models are often opaque, and data scarcity, especially in domains with emotionally rich or symbolic language.

Chapter 3

Dataset Description and Text Pre-Processing

3.1 Dataset Description

This study draws upon a rich dataset of dream narratives, originally collected via questionnaire-based protocol. Each participant (referred to as a "dreamer") was asked to describe a recent daydream, and to rate the emotional tone of their experience. These ratings, along with the narrative texts, form the foundation of this research.

Dream content was self-reported by participants using open-ended questionnaire items. Each entry consisted of a free-text dream description and a set of metadata fields capturing emotional and contextual aspects. Of particular interest were two components:

- **Dream Description:** A narrative provided by the participant, capturing the content of the daydream in natural language.
- **Emotional Rating:** A numerical score ranging from 0 to 7, where 0 indicates the most negative emotion and 7 the most positive. This self-assessed score reflects the participant's personal affective evaluation of the dream.

In total, the initial dataset contained 401 entries. However, upon inspection, many entries were found to be incomplete or unsuitable for computational analysis. Common issues included:

- Placeholder or irrelevant text (e.g., "Private," "Open-Ended Response").
- Missing or duplicated fields.
- Ambiguities in question phrasing or inconsistent formatting.

These limitations restricted the usefulness of the raw dataset for direct modeling but provided valuable insights for refining the annotation scheme and preprocessing pipeline.

3.1.1 Refined Dataset Construction

A curated dataset of 178 high-quality entries was developed by filtering out low-quality responses and structuring the sentiment annotations. Each entry in the final dataset contains:

- **Narrative Text:** Cleaned, lemmatized, and preprocessed dream descriptions.
- **Sentiment Labels:** Annotations from three different perspectives:
  - **Dreamers (7-class):** The original dream authors who rated their own emotional experience on a 0–6 scale. These labels reflect deeply personal and introspective evaluations.
  - **Independent Judges:** A group of independent annotators who rated each dream description without any background context about the dreamer. Their evaluations reflect an external, text-based interpretation of emotional tone. The judges annotated each dream using a 3-class format, with sentiment labels ranging from 0 (Negative) to 2 (Positive).
  - **Expert Annotator:** An expert annotator who applied a consistent psycholinguistic framework for sentiment evaluation, drawing on theoretical and empirical guidelines from affective computing and emotion psychology. This rater provided sentiment annotations in both the 3-class format (0–2) and a more fine-grained 7-class scale ranging from 0 (Very Negative) to 6 (Very Positive), enabling multi-resolution analysis and model evaluation.

Table 3.1 and figure 3.1 summarize the distribution of sentiment labels across the three annotator sources in both the 3-class and 7-class formats. In the 3-class setting, all three annotators, Expert Annotator, Independent Judges, and Dreamers, labeled 178 unique dream narratives. While label coverage is consistent across sources, the class distributions differ notably.

For example, the *Dreamers* rated 94 of their own dreams as positive, compared to only 53 by *Expert Annotator* and 43 by the *Independent Judges*. This suggests a positive self-bias in self-assessment, whereas the Independent Judges showed a more balanced distribution skewed slightly toward neutrality. These variations reinforce the importance of cross-source agreement analysis later in this study.

In the 7-class format, the distribution becomes even more differentiated. The *Dreamers* heavily used the upper end of the scale, whereas the *Expert Annotator*

annotations are more evenly distributed, with the highest concentration around classes 3 to 5.

These distributional discrepancies highlight the subjective nature of sentiment labeling in introspective narratives like dreams. They also emphasize the challenge of building generalizable models across annotators and the value of interpreting classifier performance in the context of human disagreement.

Table 3.1: Class distribution across annotators in both 3-class and 7-class settings.

Class Label	**Expert Annotator**	**Independent Judges**	**Dreamers**
	3-Class Setting		
0 (Negative)	39	39	48
1 (Neutral)	86	96	36
2 (Positive)	53	43	94
Total	178	178	178
	7-Class Setting		
Class Label	**Expert Annotator**	**Dreamers**	
0	7	2	
1	13	13	
2	12	22	
3	42	25	
4	47	33	
5	37	36	
6	20	47	
Total	178	178	

Rather than treating these annotations as interchangeable ground truths, this study treats each rater's perspective as a distinct lens on emotional interpretation. In later chapters, we quantitatively assess the relationships between these annotators through a series of cross-source analyses, including:

- Cross-annotator classification accuracy matrices,
- Spearman's rank correlation (ρ) for ordinal agreement,
- Cohen's kappa (κ) for categorical label alignment.

This approach allows us not only to evaluate model performance on each individual label source, but also to understand the extent to which emotional assessments are consistent, or divergent, across human perspectives. The final

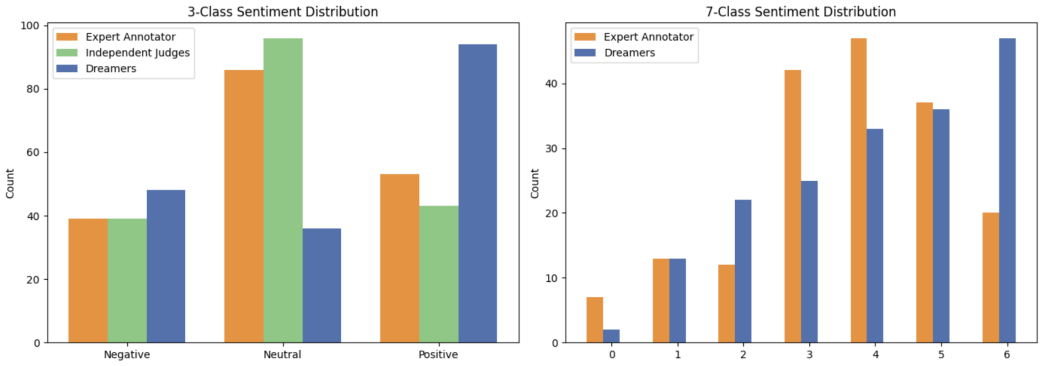


Figure 3.1: Bar charts showing sentiment class distributions for each annotator. Left: 3-class format across Dreamers (blue), Expert Annotator (orange), and Independent Judges (green). Right: 7-class format comparing Dreamers and Expert Annotator (both aligned on a 0–6 scale).

curated dataset thus serves as both a benchmark for machine learning and a case study in the subjectivity of emotional interpretation in natural language.

3.2 Text Preprocessing

Effective sentiment classification in natural language processing critically depends on the quality and consistency of the input data. Given the abstract and emotionally nuanced nature of dream narratives, a comprehensive text preprocessing pipeline was implemented to reduce noise, normalize linguistic variation, and prepare the data for robust feature extraction.

The pipeline consists of the following stages:

1. **Text Standardization:** All input text was converted to lowercase to eliminate case-based redundancy and ensure uniform tokenization. This normalization step ensures consistent treatment of terms such as *"Love"* and *"love"*.

2. **Symbol and Digit Filtering:** Using regular expressions, all non-alphabetic characters—including punctuation, numbers, and special symbols—were removed. This focuses the model on semantically meaningful tokens and prevents sparsity introduced by non-linguistic artifacts.

3. **Whitespace Normalization:** Irregular spacing and line breaks were eliminated by collapsing all whitespace into single space delimiters and trimming leading and trailing characters. This ensures consistent input formatting for token-based models.

4. **Stopword Removal:** High-frequency, low-information words (e.g., "the," "and," "in") were removed using NLTK's English stopword list. This step reduces lexical redundancy and enhances the relative weight of emotionally informative terms.

5. **Lemmatization:** Remaining words were reduced to their base forms using WordNet's lemmatizer. For example, "dreaming," "dreamed," and "dreams" were all mapped to "dream." This reduces vocabulary size and increases generalizability across morphologically related terms.

6. **Empty Document Filtering:** Any document that resulted in an empty string after preprocessing was removed from the dataset to prevent the introduction of non-informative or invalid feature vectors.

7. **Label Integrity Enforcement:** All rows with missing sentiment annotations were excluded to maintain consistency in downstream supervised learning tasks. This step ensures that every training instance has a valid label.

The result of this pipeline is a refined and semantically meaningful corpus of `cleaned_text`, which serves as the input for both TF-IDF vectorization and Empath-based semantic feature extraction. These preprocessing operations are critical to improving model interpretability, stability, and performance across noisy, subjective textual data such as dream reports.

Chapter 4

Feature Engineering

To accurately model the complex emotional and linguistic structure of dream narratives, we developed a hybrid feature extraction framework that integrates both statistical text representations and semantically enriched features. Specifically, our approach combines Term Frequency–Inverse Document Frequency (TF-IDF) vectorization with psychologically motivated semantic embeddings derived from the Empath lexicon. This dual-channel representation captures both surface-level lexical patterns, such as local n-gram frequencies, and abstract thematic dimensions rooted in emotional and cognitive categories. By jointly leveraging these complementary sources of information, the model is better equipped to recognize subtle sentiment cues and latent psychological signals expressed through natural language.

4.0.1 Empath Lexicon Features

Empath is a data-driven, lexicon-based NLP tool that maps textual content onto 194 empirically validated semantic categories, including emotionally and thematically rich dimensions such as *sadness*, *violence*, *love*, and *trust*. Built on a neural embedding model trained on 1.8 billion words of modern fiction, Empath identifies semantically related terms based on their contextual similarity in the learned vector space. It uses small sets of seed words to expand each category and validates them through crowdsourcing, ensuring human interpretability and thematic coherence. For each preprocessed dream description, Empath computes a normalized score for every category, capturing the relative frequency of category-related terms in the text. In our implementation, we set `normalize=True`, which scales the output such that each category score reflects the proportion of total matched words in the input text. Formally, for a given input sample t, the Empath analysis yields a semantic feature vector:

$$\mathbf{e}_t \in \mathbb{R}^{194},$$

where each component represents the cosine-weighted association strength between the input and a predefined category. This approach abstracts away from surface-level lexical patterns and enables a psychologically meaningful, high-level encoding of emotional and thematic structure. Compared to traditional lexicons like LIWC, Empath offers broader coverage, the ability to generate new categories on demand, and strong empirical alignment with validated psychological constructs (e.g., achieving an average Pearson correlation of $r = 0.906$ with LIWC on shared categories)[8] . These properties make Empath particularly suited for capturing latent sentiment cues embedded in dream narratives, offering interpretability, flexibility, and robustness in psychological text analysis.

4.0.2 TF-IDF Features

To complement semantic abstraction with lexical specificity, we employed Term Frequency–Inverse Document Frequency (TF-IDF), one of the most widely recognized statistical weighting schemes in text mining[9]. TF-IDF quantifies the importance of a term t in a specific document d relative to its frequency across the entire corpus D. This dual weighting captures both local relevance and global rarity, making it particularly effective in emphasizing discriminative words and suppressing overly common terms.

Formally, the TF-IDF weight is calculated as:

$$\text{TF-IDF}(t, d, D) = \text{tf}(t, d) \cdot \log\left(\frac{|D|}{|\{d_i \in D : t \in d_i\}|}\right)$$

where:

- $\text{tf}(t, d)$: frequency of term t in document d,
- $|D|$: total number of documents in the corpus,
- $|\{d_i \in D : t \in d_i\}|$: number of documents in which term t appears.

In this study, we utilized scikit-learn's implementation of `TfidfVectorizer` with the following parameters: unigrams and bigrams ($n \in \{1, 2\}$), English stopword removal, and a maximum of 1,000 features selected by highest term frequency across the corpus. This dimensionality constraint controls sparsity and computational load while retaining the most informative lexical patterns.

Each document is thus transformed into a high-dimensional, sparse feature vector $\mathbf{t}_t \in \mathbb{R}^{1000}$, where each dimension corresponds to a weighted n-gram. These vectors reflect not only word occurrence but also contextual uniqueness across the corpus, enabling the model to capture affective cues grounded in specific word usage.

As emphasized in related studies such as Hakim et al. (2014), TF-IDF has proven to be an effective approach for document classification across multiple languages and domains, offering high accuracy even in sparse textual settings. Moreover, it requires no domain-specific knowledge and is robust to noise, making it suitable for applications such as dream narrative classification where linguistic variability is high.

Finally, to ensure scale compatibility across features, the resulting TF-IDF vectors were normalized using `StandardScaler`, aligning them with the Empath-derived semantic features for unified downstream learning.

4.0.3 Feature Concatenation and Normalization

To construct a comprehensive and discriminative feature representation for each dream description, we adopted a hybrid approach that merges lexical and semantic signals into a unified vector. Specifically, we concatenated the TF-IDF-based lexical features with the Empath-derived semantic features. This combination leverages both the surface-level statistical structure of the text and its deeper psychological themes, enabling more nuanced modeling of affective patterns.

Formally, for a given input sample t, the lexical representation $\mathbf{t}_t \in \mathbb{R}^{1000}$ and the semantic representation $\mathbf{e}_t \in \mathbb{R}^{194}$ are concatenated to yield a single composite feature vector:

$$\mathbf{x}_t = [\mathbf{t}_t; \mathbf{e}_t] \in \mathbb{R}^{1194}.$$

This feature fusion strategy, commonly employed in neural architectures like multi-layer perceptrons (MLPs), enhances the network's capacity to learn joint dependencies across heterogeneous modalities. By encoding both statistical word distributions and interpretable affective categories, the model is positioned to capture a broader spectrum of patterns relevant to dream sentiment classification.

Before passing the concatenated feature vectors to the classifier, we applied standardization using the `StandardScaler` implementation, which performs z-score normalization:

$$x' = \frac{x - \mu}{\sigma},$$

where μ and σ are the mean and standard deviation of each feature, estimated exclusively from the training data.

Recent work by Amorim et al.[10] highlights that the choice of scaling technique significantly impacts classification performance, particularly in neural networks. Among the various normalization strategies evaluated, z-score normalization consistently produced stronger and more stable results across a wide range of datasets and models. The study also emphasizes that improper scaling can degrade performance more severely than no scaling at all, especially when dealing with imbalanced or

high-dimensional data. These findings support our design choice of using standard scaling in all classification pipelines.

This transformation ensures that each feature has zero mean and unit variance, preventing any one feature, especially those with larger numeric scales, from disproportionately influencing the learning process. This is particularly important in gradient-based optimization algorithms such as those used in MLPs, where imbalanced feature magnitudes can lead to unstable gradients, slow convergence, or biased weight updates.

To preserve the integrity of the evaluation process, the standardization parameters were computed strictly from the training set. These parameters (mean and standard deviation) were then applied to transform the test set. This procedure ensures that no information from the test data influences the scaling step, avoiding data leakage and ensuring that evaluation metrics reflect real-world generalization performance.

Chapter 5

Class Resampling

The sentiment label distribution in our dataset exhibited a pronounced imbalance, with certain classes containing as few as two instances while others exceeded forty. Such disproportionate representation creates a strong bias toward majority classes, leading to suboptimal decision boundaries and severely reduced predictive performance for minority categories. As highlighted in recent studies on imbalanced learning [11], this imbalance not only degrades classification accuracy but can also hinder the model's ability to generalize, as the learned hypothesis space becomes dominated by patterns from the most frequent classes.

To address the significant class imbalance in our dataset, we implemented a targeted resampling strategy based solely on SMOTE (Synthetic Minority Oversampling Technique). This approach focuses on selectively generating synthetic samples for underrepresented classes based on their original distribution. This method increases class parity without discarding potentially valuable samples or introducing excessive noise. By restoring balance through synthetic interpolation between minority instances, the model is given a more representative training signal across all sentiment categories. This design helps mitigate bias toward dominant classes while preserving the integrity and diversity of the original dataset, ultimately improving the fairness and robustness of downstream predictions.

5.0.1 Initial Class Analysis

We began by performing a quantitative assessment of the initial class distribution to identify the degree of imbalance and, in particular, to determine the size of the smallest class. This value was critical for configuring the `k_neighbors` parameter in SMOTE, as generating synthetic samples requires a sufficient number of existing instances to form a meaningful neighborhood structure in feature space. Setting this parameter too high relative to the minority class size can lead to the creation of unrealistic synthetic points, while setting it too low may fail to capture the

underlying class variability.

The initial analysis of the dataset revealed a highly uneven distribution of sentiment classes (Figure 5.1). Class frequencies ranged from a minimum of only **2 instances** in the rarest category to a maximum of **47 instances** in the most frequent one. Such disparity indicates a substantial risk of model bias toward dominant classes, as the hypothesis space would be disproportionately shaped by the patterns present in higher-frequency categories. This imbalance also implies a reduced ability to learn robust decision boundaries for underrepresented classes, thereby limiting generalization performance across the entire label space.

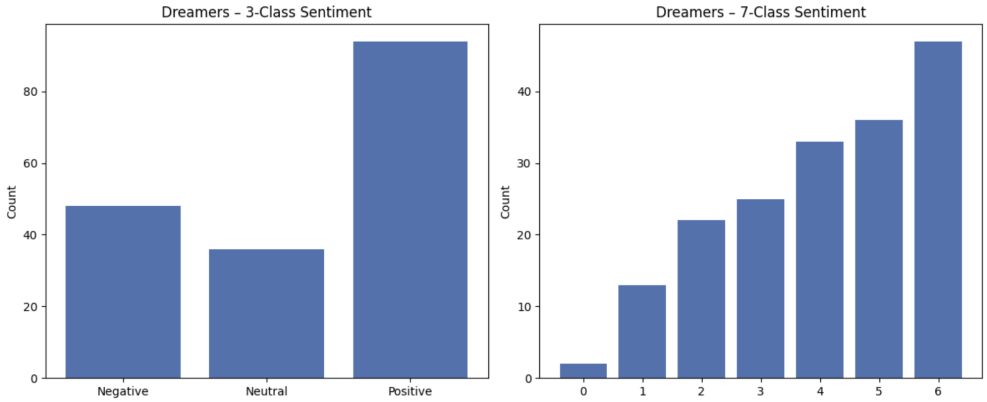


Figure 5.1: Orginal class distribution

To ensure more realistic and meaningful synthetic samples during oversampling, we applied a pre-processing step to merge the two lowest sentiment classes. Specifically, since Class 1 contained only two instances, insufficient for reliable interpolation, we combined it with Class 2, effectively treating them as a single class during resampling. This adjustment preserved the semantic proximity of the labels while allowing SMOTE to operate on a denser and more structurally coherent neighborhood. By avoiding interpolation between extremely sparse or isolated points, this step reduced the risk of generating artificial samples that do not reflect real patterns in the data, thus improving the overall quality and credibility of the augmented dataset.

5.0.2 Primary Resampling with SMOTE

To address class imbalance without discarding potentially informative data, we employed the Synthetic Minority Over-sampling Technique (SMOTE) as the sole resampling strategy. Originally proposed by Chawla et al.[12], SMOTE generates synthetic examples by interpolating between a minority class sample and one of its k nearest neighbors within the same class. This process increases the density and

continuity of the minority class region in the feature space, promoting broader and more generalizable decision boundaries.

Unlike naive oversampling via duplication, SMOTE operates directly in feature space. For a given minority sample x_i, a synthetic instance x_{new} is constructed as:

$$x_{\text{new}} = x_i + \delta \cdot (x_{nn} - x_i),$$

where x_{nn} is a randomly selected neighbor from the k nearest neighbors of x_i, and $\delta \in [0,1]$ is a randomly drawn scalar. This formulation ensures that the new sample lies along the line segment connecting x_i and x_{nn}, preserving locality while introducing novel, non-duplicated data points.

This method is particularly effective in avoiding the overfitting and fragmented decision regions that arise when replicating rare instances. By increasing the spread of minority samples throughout the feature space, SMOTE enables the classifier to learn more inclusive and smoother decision surfaces. Furthermore, because our feature set consists of high-dimensional lexical and semantic embeddings (TF-IDF and Empath), SMOTE's ability to interpolate within that space allows us to generate realistic synthetic narratives that conform to the existing distribution of emotional language patterns.

In our implementation, we applied SMOTE after all preprocessing and feature extraction steps, targeting classes with below-average frequency. The number of synthetic samples was adjusted dynamically per class to ensure controlled and interpretable balancing. For extremely sparse classes (e.g., with only one or two samples), we merged them with adjacent categories prior to resampling to avoid creating unrealistic interpolations and maintain label coherence.

This controlled application of SMOTE proved essential for producing a balanced yet semantically valid training dataset, which in turn improved the model's fairness and generalization capacity across both coarse- and fine-grained sentiment tasks.

5.0.3 Final Dataset Assembly

The final balanced dataset was constructed using SMOTE. After feature extraction and label preprocessing (including the merging of classes with insufficient support), synthetic samples were generated selectively to bring each class to an equal representation level. The final dataset was assembled by vertically stacking the original (retained) samples with the newly synthesized instances for each class.

Figure 5.2 and Figure 5.3 illustrate the class distributions before and after SMOTE-based augmentation for the 6-class and 3-class settings, respectively. In both cases, the post-resampling distribution achieves exact parity across all classes, with each class comprising an equal percentage of the total dataset.

In the 6-class case, the original distribution exhibited substantial skew, with the lowest-represented class making up only 8.4% of the data and the highest reaching

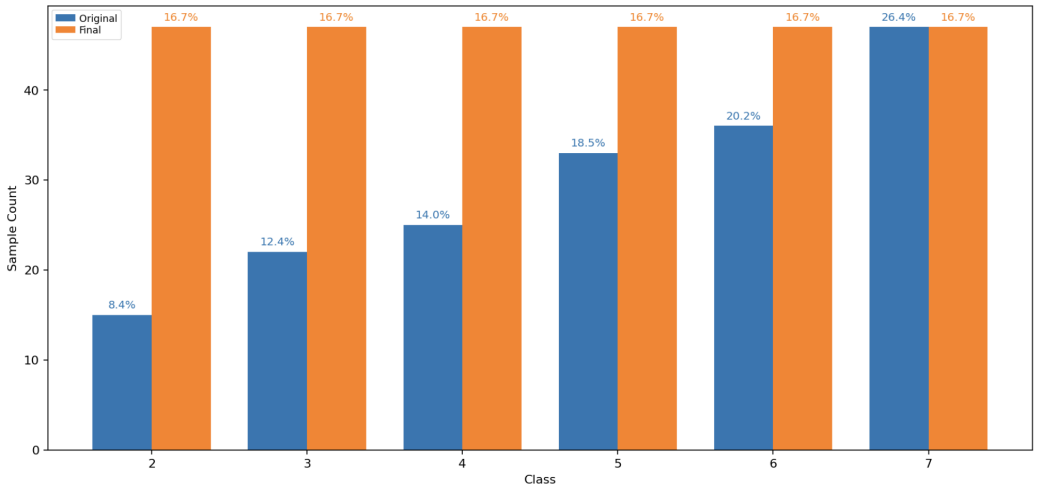


Figure 5.2: Class distribution before and after SMOTE-based resampling (6-class setting).

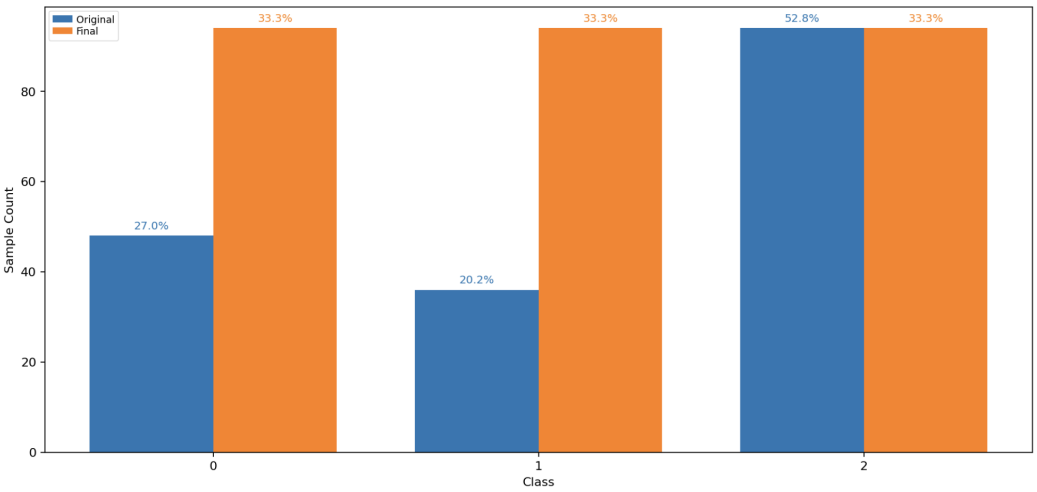


Figure 5.3: Class distribution before and after SMOTE-based resampling (3-class setting).

26.4%. After SMOTE, all six classes were balanced to 16.7%, ensuring that the model would not be disproportionately biased toward overrepresented emotional categories.

In the 3-class setting, a similar pattern emerged. The positive class originally dominated over 50% of the dataset, while the neutral and negative classes lagged significantly behind. Following augmentation, all three sentiment categories were brought to an equal share of 33.3%, making the dataset better suited for learning

fair and generalizable decision boundaries.

This balancing process was executed without excessive inflation of the dataset size or oversaturation of synthetic samples from a single class. Moreover, we applied SMOTE only after text preprocessing, class merging, and feature extraction to ensure that synthetic samples were generated within a meaningful lexical-semantic feature space, preserving the integrity of both TF-IDF and Empath representations. The resulting dataset provides a strong, uniform foundation for downstream classification, enabling the model to learn from an evenly distributed representation of the sentiment spectrum.

Chapter 6

Model Architecture

6.0.1 Model Architecture

The predictive backbone of this work is a **Multilayer Perceptron (MLP)**, a feed-forward artificial neural network composed of an input layer, one or more hidden layers, and an output layer. Each neuron computes a weighted sum of its inputs, adds a bias term, and applies a non-linear activation function. This layered non-linearity enables the network to model complex, non-linear mappings between the input feature space and the target sentiment classes.

The input layer receives the hybrid feature vector produced by our preprocessing pipeline, which concatenates lexical patterns extracted via Term Frequency–Inverse Document Frequency (TF-IDF) with semantic category scores obtained from the Empath lexicon. Before entering the network, these concatenated features are standardized using `StandardScaler` to enforce zero mean and unit variance across dimensions. As noted in[13], such normalization stabilizes gradient magnitudes during training, accelerates convergence, and reduces the risk of certain units dominating the learning process due to scale disparities.

The hidden layers in our MLP employ the `tanh` activation function, selected for its smooth, bounded output and ability to model complex non-linear relationships in the balanced, standardized feature space. Unlike `ReLU`, which can suffer from dead neuron issues, `tanh` maps inputs to the $[-1, 1]$ range, providing symmetric gradients that can improve convergence when features are zero-centered. The output layer uses the `softmax` activation, transforming the network's final raw scores into a normalized probability distribution over sentiment classes. Model training is performed using the categorical cross-entropy loss function, which is particularly effective for multi-class classification tasks, as it directly optimizes the log-likelihood of the correct class.

From a theoretical perspective, MLPs exhibit the *universal approximation property*, allowing them to approximate any continuous target function f_0 to an

arbitrary degree of accuracy given sufficient neurons and layers. However, as discussed in[14], increasing the number of hidden layers or units expands the hypothesis space, and if the model complexity significantly exceeds that of f_0, the risk of overfitting increases sharply. Overparameterized models may memorize training-specific noise rather than learning generalizable patterns.

To mitigate this risk, our architecture was deliberately configured to provide *sufficient but not excessive* capacity, large enough to capture the complexity of the hybrid lexical–semantic feature space, yet constrained to avoid unnecessary parameter growth. This balanced design, combined with input normalization, explicit regularization, data balancing strategies, and systematic hyperparameter tuning, ensured that the final model achieved strong predictive accuracy while maintaining robust generalization performance.

6.0.2 Hyperparameter Tuning

The performance of the Multi-Layer Perceptron (MLP) model was optimized using an exhaustive `GridSearchCV` procedure in `scikit-learn`. This search systematically evaluated combinations of architectural and learning parameters to identify the configuration that maximized classification performance on the balanced sentiment dataset.

The parameter grid was defined as follows:

- **Hidden layer sizes:** (100,), (200,), (200,100), (300,200,100)
- **Activation functions:** `ReLU`, `tanh`
- **Solvers:** `adam`, `sgd`
- **Regularization strength (α):** 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}
- **Learning rate schedules:** `constant`, `adaptive`
- **Maximum iterations:** 500 (fixed for all runs)

The search employed **stratified 5-fold cross-validation** to preserve class distribution within each fold. The primary selection criterion was the macro-averaged F_1-score, ensuring balanced performance across all sentiment classes regardless of frequency. Accuracy was recorded as a secondary metric.

The grid search identified the following optimal configuration:

- Hidden layers: $(200, 100)$
- Activation function: `tanh`

- Solver: `adam`
- Regularization parameter (α): 0.01
- Learning rate schedule: `constant`
- Maximum iterations: 500

This configuration achieved a **macro F_1-score of 0.7910** during cross-validation for the 3-class and **macro F_1-score of 0.7273** for the 6-class setting. The two hidden layers with progressively decreasing neuron counts allowed the network to capture complex non-linear feature interactions while reducing dimensionality in deeper layers. The `tanh` activation provided smooth, bounded outputs suitable for the balanced feature scaling applied during preprocessing. The `adam` optimizer with a constant learning rate ensured stable convergence, while the relatively high α value (0.01) acted as an effective L2 regularizer, mitigating overfitting in the high-dimensional TF-IDF + Empath feature space.

Chapter 7

Evaluation Strategy

Robust evaluation is critical to ensure that a model's performance generalizes beyond the training data and is not an artifact of sampling bias. In this study, we adopted a dual validation framework consisting of *5-Fold Cross-Validation* and *Leave-One-Out Cross-Validation (LOOCV)*. This combination balances efficiency with rigor, providing complementary insights into the generalization capacity of our Multi-Layer Perceptron (MLP) model.

7.1 5-Fold Cross-Validation

In 5-Fold Cross-Validation, the dataset is partitioned into five mutually exclusive subsets of approximately equal size. For each iteration, four folds are used for training while the remaining fold is reserved for testing. The process repeats five times, such that each fold serves once as the test set. The final performance metrics are averaged across folds, providing a stable estimate of generalization.

Mathematically, if $\mathcal{D}$ represents the dataset and $\mathcal{D}_i$ the i^{th} fold, then for each fold $i \in \{1,2,3,4,5\}$:

$$\text{Train Set} = \mathcal{D} \setminus \mathcal{D}_i, \quad \text{Test Set} = \mathcal{D}_i$$

This strategy reduces variance in performance estimation compared to a single train–test split, while maintaining computational feasibility. In our experiments, the MLP model achieved consistently high accuracy across folds as illustrated in Figure 7.1, demonstrating its ability to handle the balanced dataset produced during preprocessing. Additionally, the mean accuracy achieved was 0.7273 ($\pm$0.0511) for the 6-class Dreamers dataset and 0.7910 ($\pm$0.0429) for the 3-class Dreamers dataset, highlighting the model's robust performance across different classification complexities.

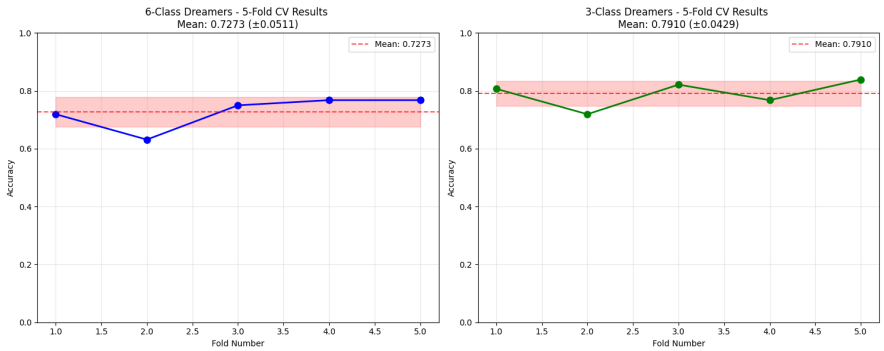


Figure 7.1: Accuracy across 5-Fold Cross-Validation.

7.2 Leave-One-Out Cross-Validation (LOOCV)

To further stress-test the model, we employed LOOCV, an extreme form of k-fold cross-validation where $k = N$ (the number of instances in the dataset). In each iteration, the model is trained on $N - 1$ samples and tested on the single remaining sample. This process is repeated N times, ensuring that every instance in the dataset is used once as a test point:

$$\text{Train Set} = \mathcal{D} \setminus \{x_i\}, \quad \text{Test Set} = \{x_i\}$$

LOOCV provides the most exhaustive utilization of data, yielding nearly unbiased performance estimates, particularly valuable in small-sample domains such as dream sentiment analysis. However, as highlighted in cross-validation literature [15], LOOCV suffers from high computational cost and higher variance compared to k-fold methods. These trade-offs were evident in our implementation: although computationally intensive, LOOCV offered granular insights into the stability of predictions across individual data points. The confusion matrices shown in Figure 7.2 further illustrate the model's performance, with the 6-class Dreamers dataset achieving an accuracy of 0.7979 and the 3-class Dreamers dataset achieving an accuracy of 0.8333, reflecting robust classification across varying complexities.

7.3 Rationale for Dual Strategy

The adoption of both 5-Fold Cross-Validation and LOOCV reflects a methodological balance between practicality and rigor. The 5-Fold approach provided a reliable, computationally efficient estimate of model performance across multiple partitions. LOOCV, though slower, served as a robustness check, ensuring that results were not contingent upon arbitrary fold assignments. Together, they form a comprehensive

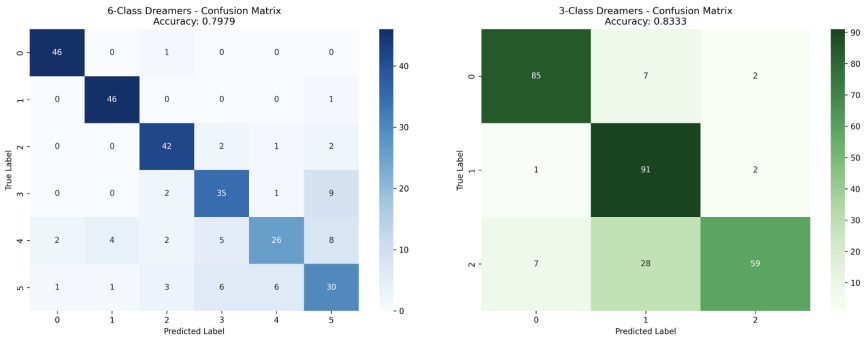


Figure 7.2: Heatmap across Leave-One-Out Cross-Validation iterations.

evaluation pipeline, aligning with best practices recommended in contemporary machine learning research [15].

Through this dual evaluation framework, we obtained both a stable estimate of model generalization (via 5-Fold CV) and a highly granular assessment of robustness (via LOOCV). These complementary strategies confirm that the proposed MLP model, trained on balanced representations of dream narratives, achieves consistent and reliable performance across a diverse set of validation conditions.

Chapter 8

Results

8.1 Classification Results by Label Granularity

8.1.1 3-Class Classification Performance

To evaluate the model's performance in detecting broad emotional tones, we mapped the original 7-point sentiment labels into three macro categories: **negative**, **neutral**, and **positive**. This 3-class setting reduces label granularity and emphasizes coarse-level emotional polarity, which is particularly suitable for downstream affective applications where interpretability and robustness are preferred over fine-grained distinctions.

The model was trained on the `Dreamers` dataset using the best-performing configuration obtained via grid search: two hidden layers with 200 and 100 neurons, `tanh` activation, `adam` solver, $\alpha = 0.01$, and a constant learning rate over 500 epochs. Evaluation was performed on a stratified 20% test split.

Table 8.1 presents the detailed classification results for each class, including precision, recall, and F_1-score, while figure 8.1 shows the corresponding confusion matrix. The model achieved a **macro-averaged F_1-score of 0.86** and an **overall accuracy of 0.86**, indicating strong and balanced performance across all categories.

Performance was consistently strong across all three classes, with particularly high scores in the *negative* and *positive* classes. Slightly lower metrics for the *neutral* class are consistent with its semantic ambiguity and higher overlap with adjacent classes. Nonetheless, the model exhibited no significant signs of bias toward any single class, confirming that the class-balancing techniques (SMOTEENN and controlled duplication) were effective.

These results validate the effectiveness of the hybrid TF-IDF and Empath feature set in capturing the emotional and semantic structure of dream narratives, and demonstrate that even a moderately deep MLP can yield high-quality, generalizable predictions in the coarse-grained sentiment setting.

Table 8.1: Classification report for 3-class sentiment prediction (Dreamers).

Class	Precision	Recall	F1-Score
Negative	0.88	0.89	0.89
Neutral	0.82	0.83	0.82
Positive	0.87	0.87	0.87
Macro Avg.	**0.86**	**0.86**	**0.86**
Accuracy		**0.86**	

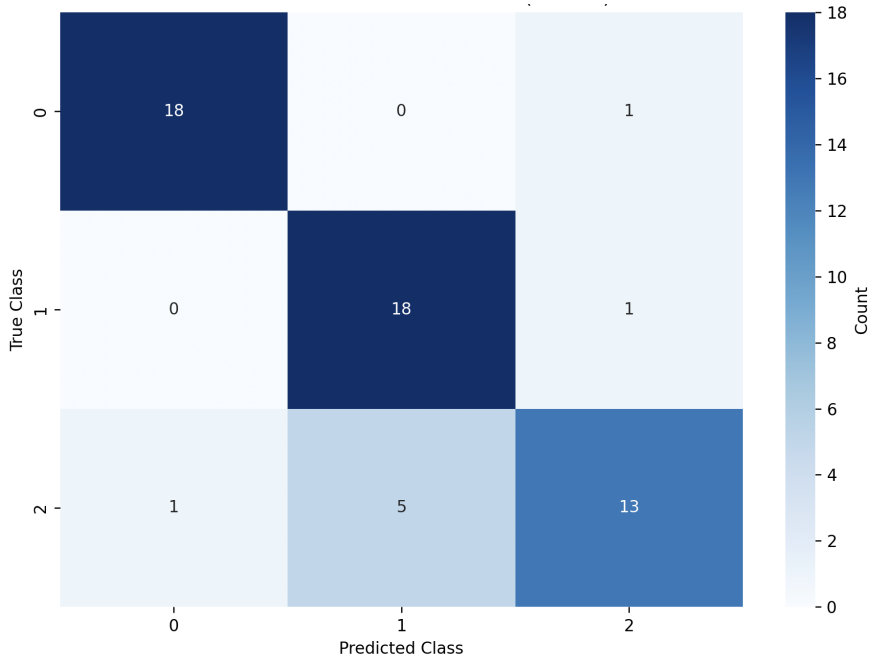


Figure 8.1: Confusion Matrix when trained and tested on Dreamers (3-class)

8.1.2 6-Class Classification Performance

To evaluate the model's ability to detect more nuanced emotional tones, we conducted a second experiment using a 6-class version of the `Dreamers'` dataset. This setting retains finer granularity across the sentiment scale while excluding only the rarest class (original class 7), which was underrepresented and removed during preprocessing due to insufficient support.

This classification task is more challenging than the 3-class version due to the increased number of classes and finer inter-class distinctions. The model configuration remained identical to that used in the 3-class experiment: two hidden layers with 200 and 100 neurons, `tanh` activation, `adam` solver, $\alpha = 0.01$, and 500

training iterations. The same stratified 80/20 train-test split was used to maintain consistency in evaluation.

Table 8.2 reports the precision, recall, and F_1-score for each of the six sentiment classes, along with the macro-averaged scores and overall accuracy and figure 8.2 shows the corresponding confusion matrix.

Table 8.2: Classification report for 6-class sentiment prediction (Dreamers).

Class	**Precision**	**Recall**	**F1-Score**
Class 1 (Very Negative)	0.89	0.83	0.86
Class 2 (Negative)	0.77	0.89	0.83
Class 3 (Slightly Negative)	0.79	0.83	0.81
Class 4 (Slightly Positive)	0.83	0.79	0.81
Class 5 (Positive)	0.80	0.75	0.77
Class 6 (Very Positive)	0.88	0.83	0.85
Macro Avg.	**0.83**	**0.82**	**0.82**
Accuracy		**0.81**	

The model achieved a **macro-averaged F_1-score of 0.82** and an **accuracy of 0.81**, showing only a modest decrease in performance compared to the 3-class task. This drop is expected given the increased complexity of the label space and the relatively lower sample sizes per class.

Notably, the model performed best on the extreme sentiment classes (Class 1 and Class 6), which are more distinct in language and emotional polarity. Performance was slightly lower for intermediate classes (Class 3, 4, and 5), where semantic and emotional overlap is higher. This observation aligns with prior literature on class confusion in fine-grained sentiment classification tasks.

These results suggest that while the MLP generalizes well across a more detailed label space, further performance improvements may require additional feature engineering, ensemble methods, or label smoothing strategies to reduce boundary confusion between adjacent classes.

8.1.3 Cross-Source Evaluation: Annotator Generalization (3-Class)

To assess the model's ability to generalize across labeling sources, we conducted a series of cross-source experiments in the 3-class setting. In this setup, models were trained on labels provided by one source (e.g., Expert Annotator, Independent Judges, Dreamers) and tested on another, allowing us to evaluate annotation

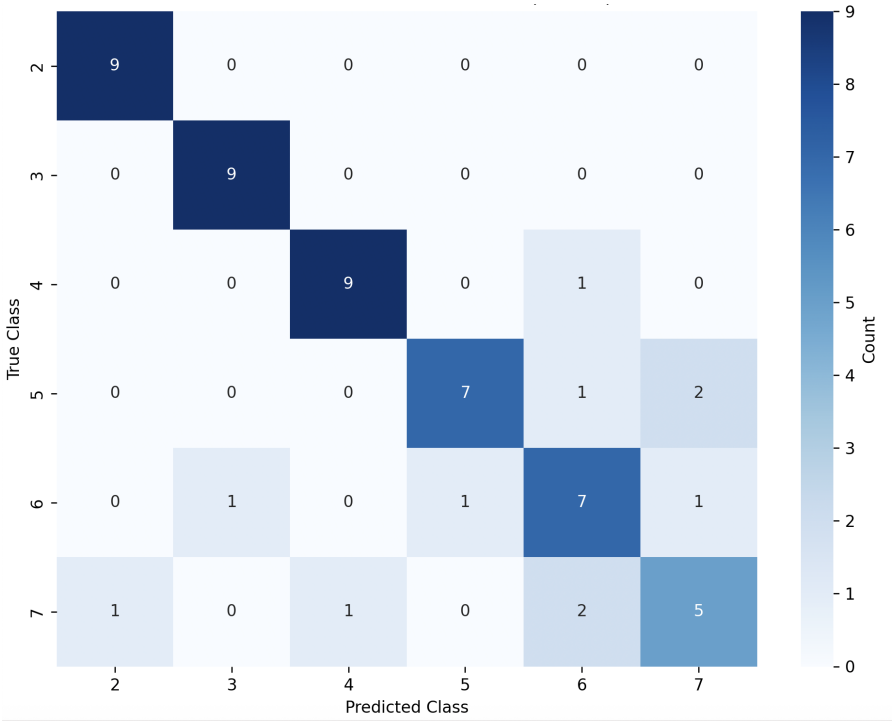


Figure 8.2: Confusion Matrix when trained and tested on Dreamers (6-class)

consistency and the transferability of sentiment representations across human annotators.

Figure 8.3 summarizes the results of this cross-source evaluation, where each cell reports the classification accuracy achieved when the model was trained on the row source and tested on the column source. Detailed performance metrics, including precision, recall, and F1-score matrices, are reported in figures 8.4, 8.5 and 8.6 respectively.

As expected, models trained and tested on the same annotator set (diagonal values) achieved the highest accuracy, particularly for the Independent Judges (91.38%) and Dreamers (85.96%) labels. However, when evaluated across sources, performance dropped substantially, especially for the Dreamers model tested on Expert Annotator (38.89%) and Judges (36.11%) labels.

These discrepancies reveal a considerable divergence in how different annotator groups interpret and label emotional tone in dream narratives. Models trained on one annotator set often fail to generalize to another, reflecting both subjective annotation practices and potential inconsistency in sentiment granularity or definition across groups. This supports previous findings that sentiment annotation—especially in abstract domains like dreams, is highly subjective and source-dependent.

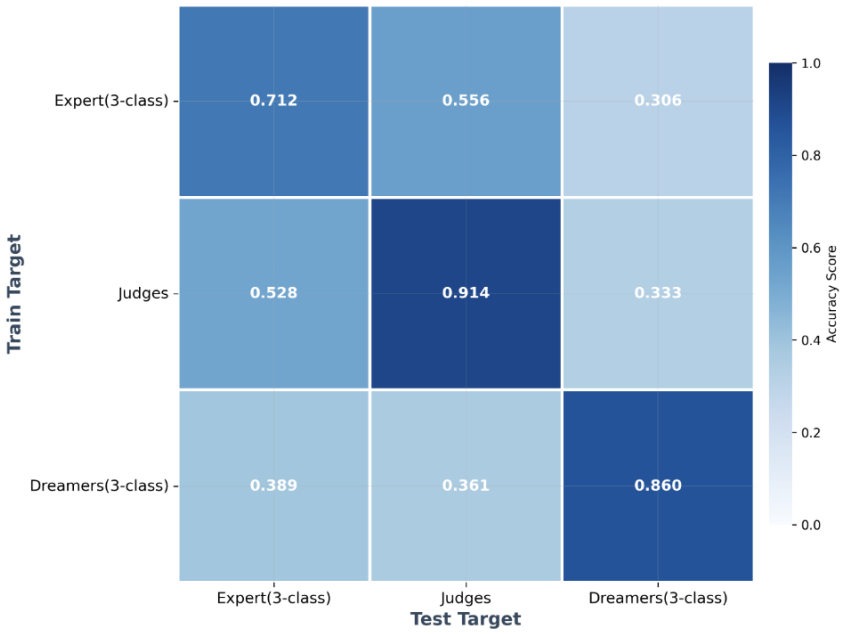


Figure 8.3: 3-Class cross-source **accuracy matrix** (Train → Test).

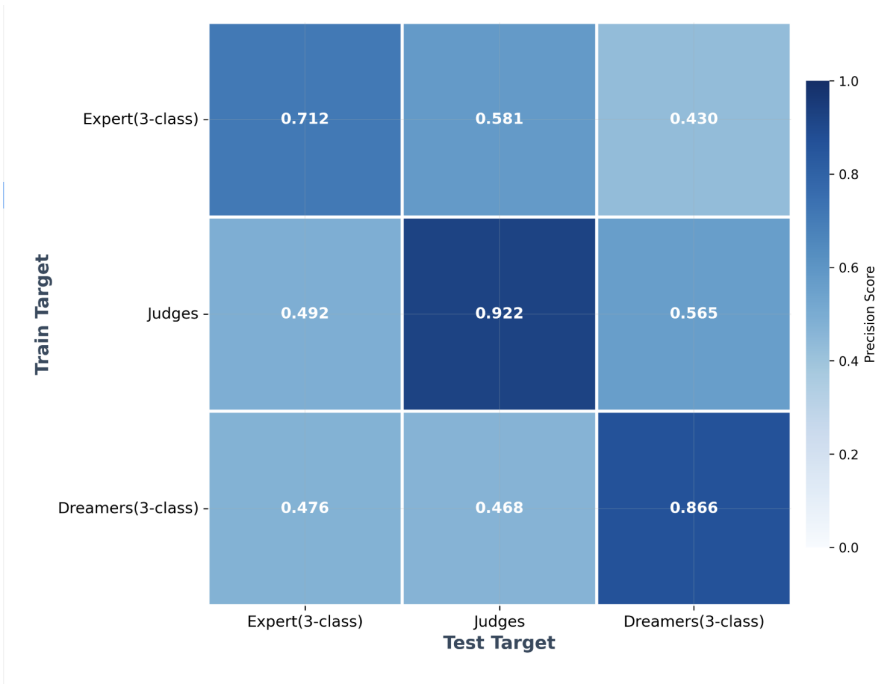


Figure 8.4: 3-Class cross-source **precision** matrix

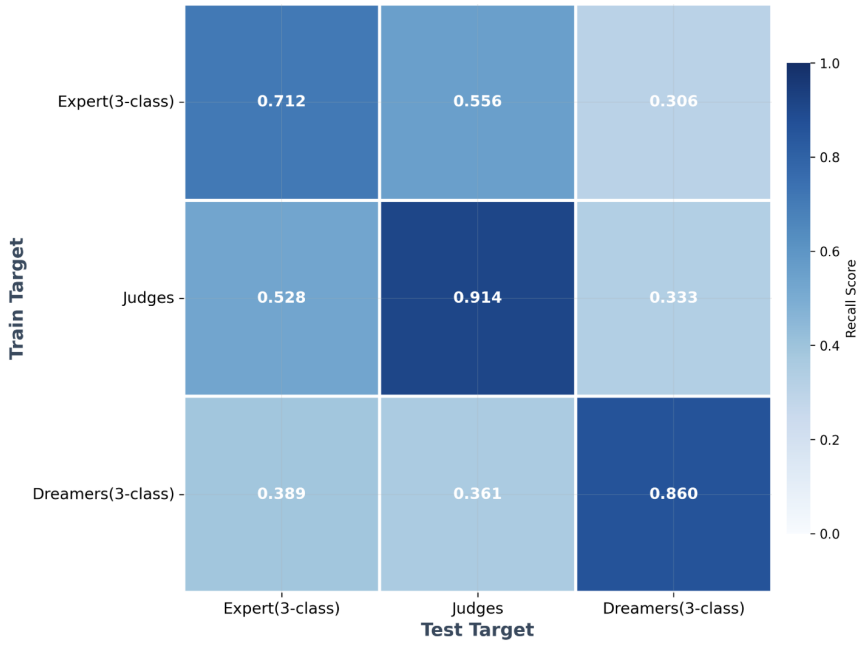


Figure 8.5: 3-Class cross-source **recall** matrix (3-class)

While these results highlight the limitations of training on a single annotator perspective, they also underscore the importance of label alignment, multi-annotator fusion techniques, or domain adaptation approaches when deploying such models across diverse populations or applications.

8.1.4 Cross-Source Evaluation: Annotator Generalization (6-Class)

To further evaluate the generalization ability of the model across annotators, we repeated the cross-source classification experiments in the 6-class setting. In this setup, the sentiment label space is more granular, making the task of cross-source learning even more challenging due to greater annotation ambiguity and finer distinctions between classes.

Figure 8.7 presents the resulting 2x2 heatmap, where each cell indicates the test accuracy achieved when the model is trained on one annotator source (row) and tested on another (column). The two annotator groups included in this analysis are the *Dreamers (6-class)* and *Expert Annotator (6-class)*, with class mappings aligned to the common 6-class structure. Detailed performance metrics, including precision, recall, and F1-score matrices, are reported in figures 8.8, 8.9 and 8.10 respectively.

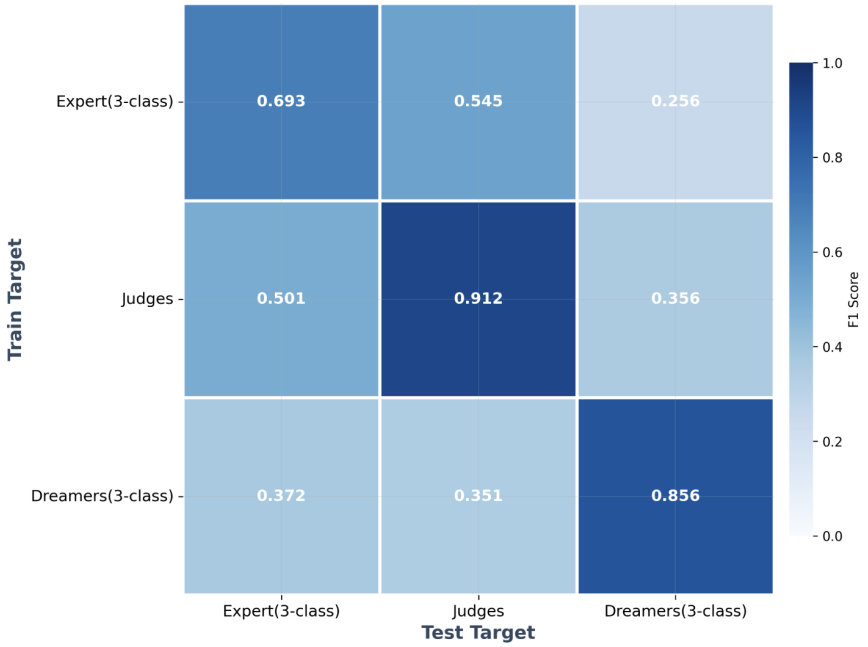


Figure 8.6: 3-Class cross-source **F1-score** matrix

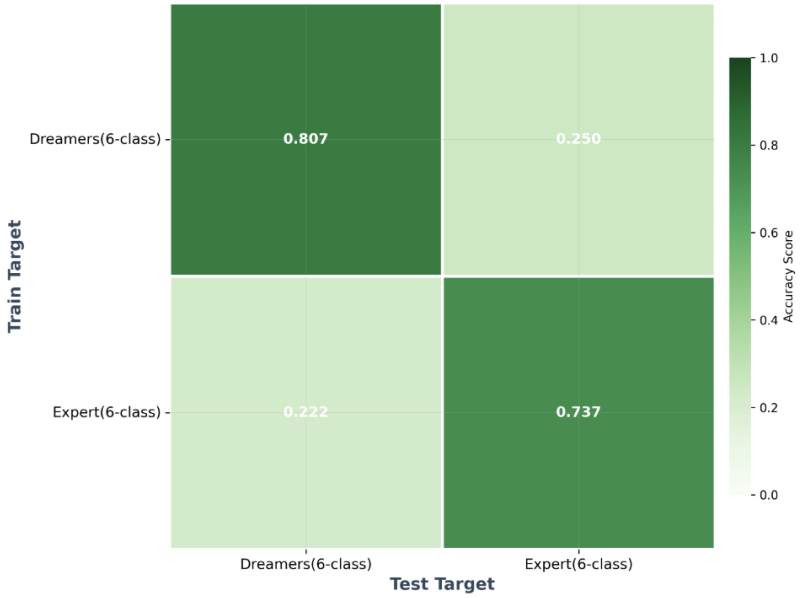


Figure 8.7: 6-Class cross-source **accuracy matrix** (Train → Test).

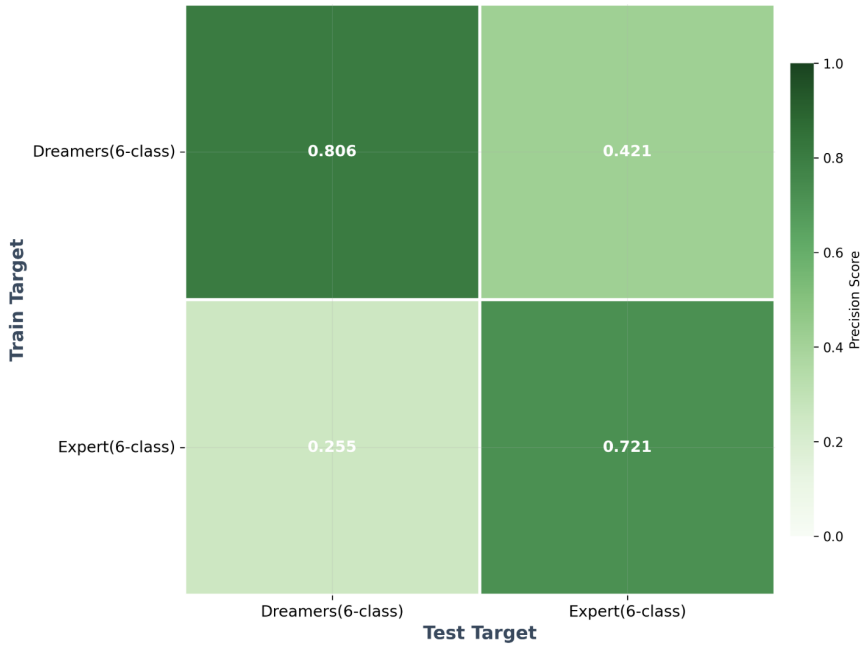


Figure 8.8: 6-Class cross-source **precision** matrix

The results indicate that intra-source accuracy remains high, **80.7%** for *Dreamers* → *Dreamers* and **73.68%** for *Expert Annotator* → *Expert Annotator*. However, cross-source generalization drops sharply: training on *Expert Annotator* and testing on *Dreamers* results in only **22.22%** accuracy, while the reverse (*Dreamers* → *Expert Annotator*) yields **25.00%**.

This significant drop illustrates the challenge of transferring fine-grained sentiment understanding between annotator groups in abstract, introspective domains such as dream narratives. The results suggest that label boundaries and class interpretations vary not only in polarity but also in subtle gradations of intensity. Unlike the 3-class case, where coarse emotional tone aligns more consistently across annotators, the 6-class structure appears more susceptible to inter-annotator divergence.

These findings highlight the need for further investigation into inter-annotator agreement, possible class relabeling strategies, and the development of models that can accommodate soft or probabilistic label alignments across sources.

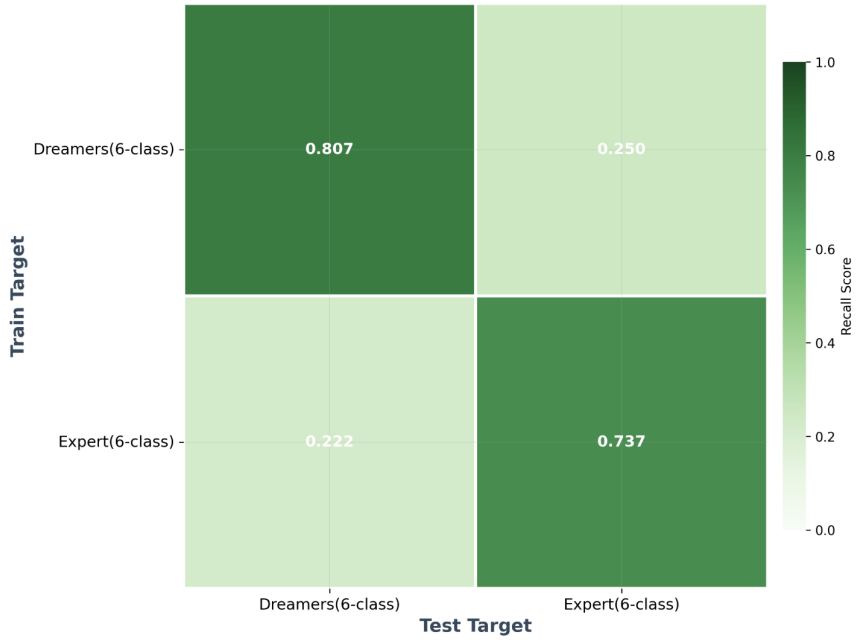


Figure 8.9: 6-Class cross-source **recall** matrix (3-class)

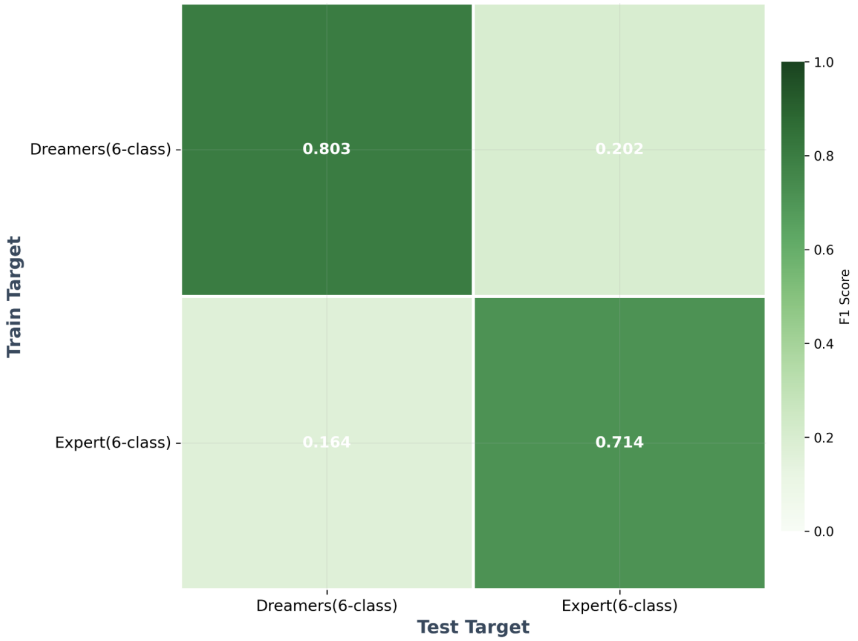


Figure 8.10: 6-Class cross-source **F1-score** matrix

8.1.5 Correlation Analysis as Baseline for Cross-Source Accuracy

To better interpret the cross-source accuracy results presented in the previous section, we introduced a correlation-based baseline analysis. While accuracy reflects model performance when transferring across annotator domains, it does not directly inform us about how similar the annotators themselves are in their labeling behavior. Without such context, it's difficult to determine whether a drop in cross-source accuracy is due to poor model generalization or inherent disagreement between annotators.

To address this, we computed pairwise correlation metrics between annotator label sets prior to model training. These correlations serve as a theoretical ceiling or lower-bound expectation for cross-source model performance. If two annotators are highly correlated in how they label the same data, we would reasonably expect a model trained on one to perform well when tested on the other. Conversely, if two annotators demonstrate low or no correlation, even a well-performing model may struggle to generalize across that boundary.

This approach frames model performance not only in terms of algorithmic success but also in relation to human disagreement and subjective interpretation—particularly important in sentiment tasks involving introspective content such as dream reports.

To operationalize this idea, we employed two widely used inter-rater agreement metrics:

- **Spearman's Rank Correlation Coefficient** – to quantify the monotonic relationship between ordinal sentiment scores across annotators.

- **Cohen's Kappa Coefficient** – to estimate the categorical agreement beyond chance, reflecting how often two annotators assign the same label to the same item.

These correlations were computed for both the 3-class and 6-class settings, and the results are visualized using cross-annotator heatmaps. Comparing these correlation matrices with the corresponding cross-source accuracy matrices allows us to distinguish model generalization failures from fundamental labeling inconsistencies between sources.

In the following sections, we present and analyze these correlations in detail, interpreting how human annotation alignment corresponds, or fails to correspond, to model transferability across annotation domains.

8.1.6 Spearman's Rank Correlation as an Agreement Baseline

To contextualize cross-source accuracy, we first quantify how similarly annotators order the same items. We use *Spearman's rank correlation* (r_s)[16], a nonparametric measure of monotonic association between two ordinal variables. It provides an agreement baseline: if two annotators rank items similarly (high r_s), a model trained on one is expected to transfer better to the other; when r_s is low, poor cross-source accuracy is expected even for a strong classifier.

Definition and computation. Let $\{(x_i, y_i)\}_{i=1}^n$ be paired labels for the same n items from two annotators. Rank each series (average ranks for ties), obtaining $R_i = \text{rank}(x_i)$ and $S_i = \text{rank}(y_i)$.

$$r_s \;=\; \frac{\sum_{i=1}^n (R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_{i=1}^n (R_i - \bar{R})^2}\sqrt{\sum_{i=1}^n (S_i - \bar{S})^2}}$$

Equivalently, when there are *no ties*, $d_i = R_i - S_i$ and

$$r_s \;=\; 1 - \frac{6\sum_{i=1}^n d_i^2}{n(n^2-1)}.$$

We use the rank-covariance form (first equation) with average ranks to handle ties robustly. For large n, significance can be approximated via the usual Fisher/Student transforms on r_s, but here we focus on effect sizes as agreement baselines.

8.1.7 Results (3-class).

Figure 8.11 shows the 3-class Spearman matrix. We observe strong rank agreement between *Expert annotator* and *Independent Judges* ($r_s = 0.885$), and much weaker agreement between *Dreamers* and either *Expert Annotator* ($r_s = 0.260$) or *Judges* ($r_s = 0.237$). These values align with the cross-accuracy matrix: moderate transfer between *Expert Annotator*→*Independent Judges* and *Independet Judges*→*Expert Annotator* ($\approx 0.53-0.56$), but markedly lower transfer when *Dreamers* is involved ($\approx 0.30-0.39$). Thus, drops in cross-source accuracy reflect genuine inter-annotator disagreement rather than model failure.

8.1.8 Results (6-class).

With finer labels (6-class), agreement diminishes further: *Dreamers* vs. *Expert Annotator* yields $r_s = 0.164$ (Figure 8.12). This very low monotonic association

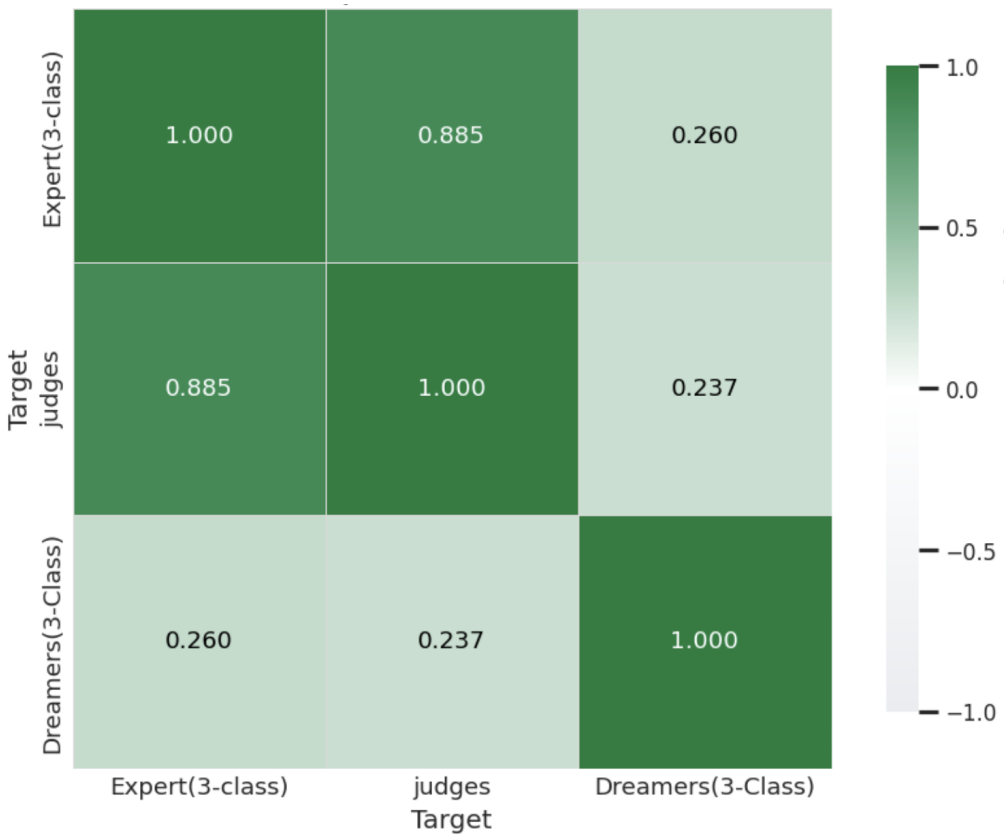


Figure 8.11: Spearman rank correlation (3-class).

matches the sharp collapse in cross-source accuracy (≈ 0.22–0.25), indicating that annotators apply substantially different ordinal criteria at higher granularity.

Overall, r_s tracks the transferability trend: higher rank agreement $\Rightarrow$ higher cross-source accuracy; low agreement $\Rightarrow$ limited transfer. Practically, these results suggest (i) harmonizing label definitions or using multi-annotator fusion when training cross-domain models, and (ii) treating r_s as a principled upper-bound indicator for expected cross-accuracy between annotator sources.

8.1.9 Cohen's Kappa Agreement (3-Class Setting)

While Spearman's correlation captures ordinal alignment, it does not evaluate exact label agreement. To measure inter-annotator consistency at the categorical level, we employed **Cohen's kappa** (κ), which quantifies observed agreement while adjusting for chance.

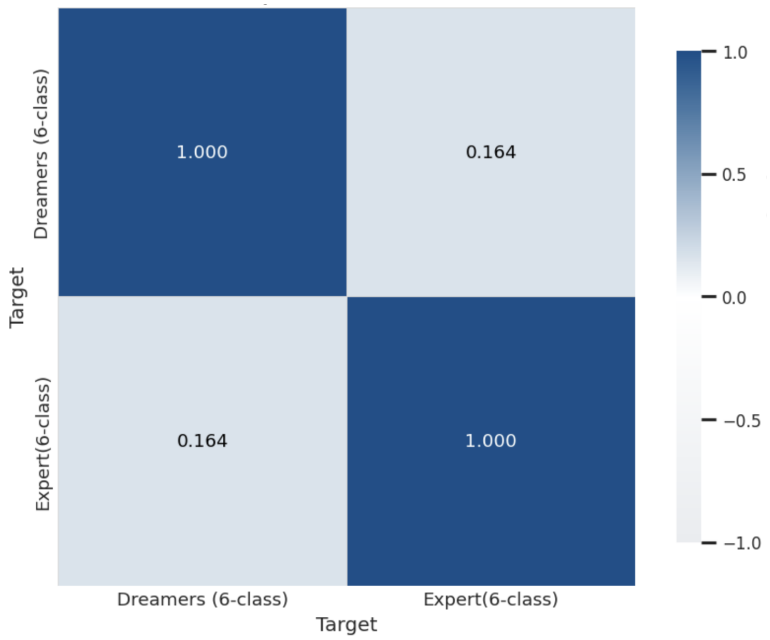


Figure 8.12: Spearman rank correlation (6-class).

Definition. Cohen's kappa is defined as:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

where:

- P_o is the observed agreement (i.e., proportion of instances where two annotators assigned the same class),
- P_e is the expected agreement by chance, computed from the product of marginal probabilities.

This correction accounts for situations in which annotators might match labels by coincidence rather than shared understanding. The statistic ranges from -1 to $+1$, where $\kappa = 1$ implies perfect agreement, $\kappa = 0$ suggests agreement no better than random, and negative values indicate systematic disagreement.

According to McHugh (2012)[17], the following guidelines offer a practical interpretation:

- $\kappa < 0.20$: None to slight agreement
- $0.21 \leq \kappa < 0.40$: Minimal agreement

- $0.41 \leq \kappa < 0.60$: Weak agreement
- $0.61 \leq \kappa < 0.79$: Moderate agreement
- $0.80 \leq \kappa < 0.90$: Strong agreement
- $\kappa > 0.90$: Almost perfect agreement

Figure 8.13 shows the kappa matrix for the 3-class setting. We observe the following pairwise values:

- **Expert Annotator vs. Independent Judges:** $\kappa = 0.846$ – strong agreement
- **Expert Annotator vs. Dreamers:** $\kappa = 0.099$ – slight agreement
- **Independent Judges vs. Dreamers:** $\kappa = 0.059$ – no meaningful agreement

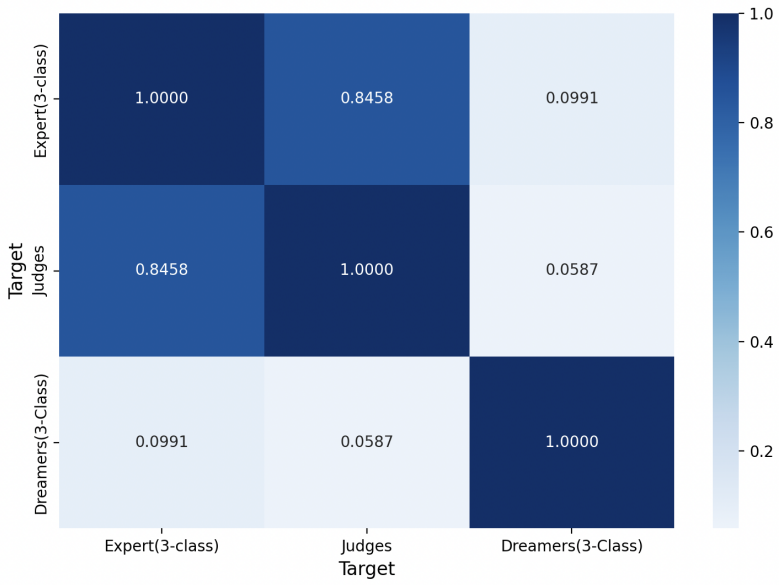


Figure 8.13: Cohen's kappa matrix for 3-class annotation alignment.

These results mirror our earlier findings based on Spearman's r_s and cross-source accuracy. The strong agreement between *Expert Annotator* and *Independent Judges* explains why models trained on one generalize reasonably well to the other. Conversely, the near-zero kappa values involving *Dreamers* suggest a mismatch in how annotators conceptualize emotional categories, either due to subjective interpretation, inconsistent labeling, or different rating criteria altogether.

Unlike percent agreement, Cohen's kappa penalizes "false agreement" that may arise by chance. Therefore, the very low κ values between Dreamers and the other

two sources are not merely indicators of inconsistency but statistically meaningful evidence of divergent labeling schemes.

Cohen's kappa provides a robust, categorical-level agreement baseline. When used alongside Spearman's ordinal measure, it offers a more complete picture of annotation quality. Together, these metrics justify our decision to analyze annotator-specific generalization separately and reinforce the role of subjective divergence as a limiting factor in cross-source model transferability.

Chapter 9

Exploring DeBERTa with Parameter-Efficient Fine-Tuning

Transformer-based pre-trained language models have become the backbone of modern natural language processing (NLP), offering remarkable gains in a wide range of tasks, including sentiment classification. Among them, **DeBERTa** (Decoding-enhanced BERT with Disentangled Attention) introduces architectural improvements that lead to both better performance and higher efficiency compared to BERT and RoBERTa [18].

Unlike conventional transformers that combine token content and positional embeddings via simple summation, DeBERTa disentangles these two components and processes them separately. This allows the attention mechanism to model richer interactions between token content and position. The architecture also includes an *Enhanced Mask Decoder (EMD)*, which reintroduces positional information only during decoding, reducing noise during the encoding phase.

DeBERTa achieves state-of-the-art results on multiple benchmarks such as MNLI, SQuAD, and SuperGLUE, while being more parameter-efficient and semantically aware. These properties make it especially suitable for nuanced tasks such as sentiment analysis of introspective and emotionally rich narrative texts, like the dream descriptions studied in this thesis.

To fine-tune DeBERTa for our specific domain with limited computational resources, we adopt **LoRA** (Low-Rank Adaptation), a parameter-efficient tuning method. LoRA allows adaptation by injecting trainable rank-decomposed matrices into the attention layers while keeping the pre-trained weights frozen. This significantly reduces the number of trainable parameters and memory consumption, making fine-tuning feasible without sacrificing model performance.

The rest of this chapter introduces the architecture of DeBERTa in more detail, explains the rationale for LoRA-based fine-tuning, and presents the results of adapting this architecture to our dataset.

9.0.1 DeBERTa Model Architecture

DeBERTa enhances the transformer encoder by introducing two core innovations: the *Disentangled Attention Mechanism* and the *Enhanced Mask Decoder*. Together, they improve contextual representation, reduce positional interference, and enable efficient transfer to downstream tasks.

Disentangled Attention Mechanism. In BERT and most transformers, token content and absolute position embeddings are added together before attention. In contrast, DeBERTa maintains separate embeddings for content and position and computes attention using multiple projections:

$$\text{Attention}_{i,j} \propto (Q_i^c)^T K_j^c + (Q_i^c)^T K_{(i-j)}^r + (Q_i^r)^T K_j^c$$

Where:

- Q_i^c, K_j^c: content query and key
- Q_i^r, $K_{(i-j)}^r$: relative position query and key

This disentangled structure enables richer modeling of semantic and structural relationships, leading to more expressive contextual embeddings.

Enhanced Mask Decoder (EMD). In BERT, absolute position embeddings are introduced early in the encoder. DeBERTa removes them from the encoder entirely and instead introduces absolute positions only during decoding through the Enhanced Mask Decoder. This reduces the burden of position modeling during context learning and improves generalization to downstream tasks.

Training Optimizations. DeBERTa also incorporates pre-activation LayerNorm, which stabilizes training, and supports both Masked Language Modeling (MLM) and Replacement Token Detection (RTD) objectives. The model is released in multiple sizes (base, large, v3 variants) with varying numbers of layers and hidden dimensions.

Table 9.1: DeBERTa model variants (source: He et al., 2021).

Model	Parameters	Layers	Hidden Size	Heads
DeBERTa Base	139M	12	768	12
DeBERTa Large	386M	24	1024	16
DeBERTa V3 Base	184M	12	768	12

In this study, we adopt the **DeBERTa v3 Base** model for its balance of efficiency and performance, integrating it with LoRA for domain-specific fine-tuning on dream sentiment classification.

9.0.2 Parameter-Efficient Fine-Tuning with LoRA

As modern transformer-based language models grow increasingly large, full fine-tuning becomes computationally and memory-wise prohibitive. To address this limitation, we adopt **Low-Rank Adaptation (LoRA)** [19], a parameter-efficient fine-tuning method that significantly reduces the number of trainable parameters while preserving or even improving task performance.

As illustrated in Figure 9.1, LoRA enables adaptation by training only low-rank matrices A and B, while keeping the main model weights frozen. This reparameterization significantly reduces the number of trainable parameters and enables lightweight fine-tuning. Furthermore, task-switching becomes trivial—only the small A, B matrices need to be swapped, which is especially beneficial for scenarios where multiple downstream tasks must share a common foundation model.

In fact, rather than updating all parameters of the pre-trained model, LoRA keeps the original weights $\mathbf{W}_0 \in \mathbb{R}^{d\times d}$ frozen and injects trainable low-rank matrices $\mathbf{A} \in \mathbb{R}^{r\times d}$ and $\mathbf{B} \in \mathbb{R}^{d\times r}$ into the forward pass as an additive reparameterization:

$$\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W} = \mathbf{W}_0 + \mathbf{BA}$$

where $r \ll d$, allowing efficient adaptation using only a small number of parameters. During training, gradients are computed and applied solely to $\mathbf{A}$ and $\mathbf{B}$, keeping $\mathbf{W}_0$ intact.

LoRA is motivated by the observation that the changes induced by full fine-tuning lie in a low intrinsic rank subspace. Thus, instead of full updates, low-rank decompositions suffice to capture task-specific knowledge. This brings multiple advantages:

- **Storage Efficiency:** Only a small set of low-rank weights need to be stored per task.

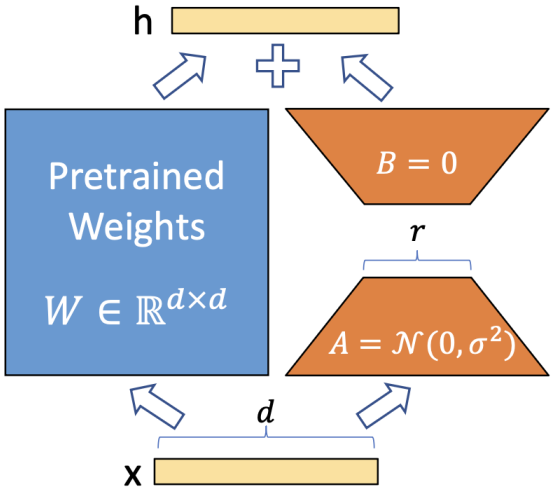


Figure 9.1: Low-Rank Adaptation (LoRA) reparameterization approach. Instead of updating the full pre-trained weight matrix $W \in \mathbb{R}^{d \times d}$, LoRA introduces two trainable low-rank matrices $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{d \times r}$. During fine-tuning, the pre-trained weights are kept frozen, and only the low-rank update $\Delta W = BA$ is trained. This design allows efficient storage, fast task switching, and reduced memory overhead. (Adapted from [19])

- **Compute Efficiency:** Memory and computation are saved by avoiding gradients on the full model.

- **No Inference Overhead:** At inference time, the low-rank update can be merged into the base weights, maintaining the same computational cost as the original model.

- **Task Modularity:** Enables rapid task-switching by swapping only the LoRA modules.

LoRA is typically applied to key weights in the attention mechanism of transformers. For a self-attention layer with projection matrices $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$, we apply LoRA to the query and value projections $(\mathbf{W}_q, \mathbf{W}_v)$. This choice strikes a balance between expressiveness and parameter efficiency [19].

In our implementation, we fine-tuned DeBERTa using LoRA with different configurations depending on the classification setting.

9.0.3 Training Configuration

We fine-tuned `microsoft/deberta-v3-base`, with a maximum sequence length of 256 tokens.

To provide a clear overview of the model setup, the following tables summarize the most important elements of the DeBERTa training pipeline. Table 9.2 outlines the LoRA configuration used for parameter-efficient fine-tuning. Table 9.3 details the training schedule and hyperparameters. Finally, Table 9.4 presents the data augmentation strategy used to expand and diversify the input space for improved generalization.

Table 9.2: LoRA Configuration

Parameter	**Value**
LoRA Rank (r)	32
LoRA Scaling (α)	64
Dropout Rate	0.1

Table 9.3: Training Schedule

Parameter	**Value**
Epochs	15
Batch Size (Train/Eval)	8
Learning Rate	1×10^{-5}
Warmup Steps	100
Weight Decay	0.01
Train/Test Split	80/20 (Stratified)
Class Balancing	Weighted loss

Table 9.4: Data Augmentation Summary

Aspect	**Details**
General Augmentation	2×
Minority Class Boost	2×
Techniques Applied	1. Synonym Replacement (dream-related terms) 2. Paraphrase Augmentation 3. Random Insertion 4. Random Deletion (5–15% of tokens)

9.0.4 DeBERTa Fine-Tuning Performance on Dreamers (3-Class Setting)

To assess how well the fine-tuned DeBERTa model captures subjective emotional patterns expressed by the dreamers themselves, we trained a dedicated 3-class classifier exclusively on the Dreamers' self-annotated dataset. The objective was to measure the model's capacity to internalize introspective emotional cues, where sentiment labels ranged from 0 (negative) to 2 (positive).

Table 9.5 summarizes the full classification metrics across all sentiment categories, while Figure 9.2 visualizes the corresponding confusion matrix.

Table 9.5: Full Classification Report – 3-Class Dreamers Model

Class	Precision	Recall	F1-score	Support
0	0.6667	0.2500	0.3636	8
1	0.0000	0.0000	0.0000	5
2	0.5833	1.0000	0.7368	14
Accuracy			0.5926	27
Macro avg	0.4167	0.4167	0.3668	27
Weighted avg	0.5000	0.5926	0.4898	27

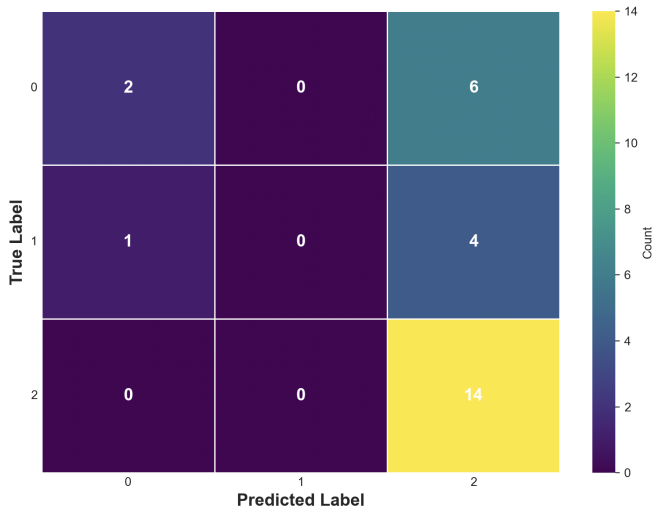


Figure 9.2: Confusion matrix for the fine-tuned DeBERTa model on Dreamers' 3-class sentiment labels.

The model achieved an overall accuracy of **59.3%**, with substantial variation across classes. Class 2 (positive sentiment) was predicted most reliably, achieving

an F1-score of **0.74** and perfect recall (**1.00**), suggesting the model effectively captures the lexical and contextual cues associated with positive emotional content.

Conversely, the model struggled with neutral and negative sentiments, particularly Class 1, which exhibited no successful predictions. This discrepancy may reflect both the limited sample size and the subjective inconsistency in mid-range (neutral) labeling by dreamers, where emotional ambiguity is high and linguistic markers are subtle.

The confusion matrix (Figure 9.2) illustrates a strong directional bias toward positive classification, confirming that while DeBERTa successfully identifies positive affect, it tends to overgeneralize ambiguous or weakly negative expressions as positive. This aligns with findings in prior affective computing literature, where transformer-based models fine-tuned on introspective text often overfit to dominant sentiment categories due to the low inter-class separation of subjective emotions.

9.0.5 DeBERTa Fine-Tuning Performance on Dreamers (6-Class Setting)

To further test the model's sensitivity to finer-grained emotional distinctions, the DeBERTa model was fine-tuned on the Dreamers' self-annotated dataset using the 6-class labeling scheme. This version expands the sentiment resolution, ranging from 2 (strongly negative) to 7 (strongly positive), and was designed to assess whether transformer-based contextual encoders can capture subtle emotional gradations within introspective narratives.

The fine-tuning configuration followed the same architecture as the 3-class experiment—combining LoRA-based parameter-efficient adaptation, class weighting, and mild data augmentation, while keeping the same number of epochs and learning rate for comparability. The classification metrics and confusion matrix are reported in Table 9.6 and Figure 9.3 respectively.

Table 9.6: Full Classification Report – 6-Class Dreamers Model

Class	Precision	Recall	F1-score	Support
2	0.0000	0.0000	0.0000	2
3	0.0000	0.0000	0.0000	3
4	0.0000	0.0000	0.0000	4
5	0.0000	0.0000	0.0000	5
6	0.2222	1.0000	0.3636	6
7	0.0000	0.0000	0.0000	7
Accuracy			0.2037	27
Macro avg	0.0370	0.1667	0.0606	27
Weighted avg	0.0494	0.2222	0.0808	27

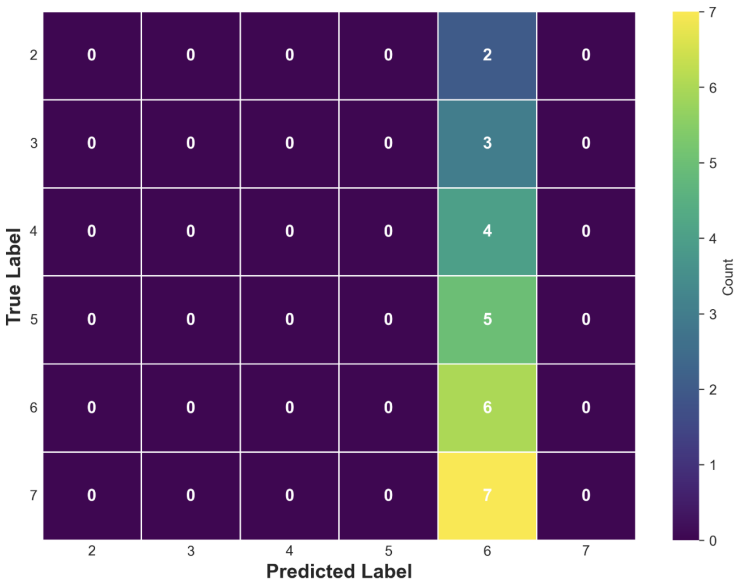


Figure 9.3: Confusion matrix for the fine-tuned DeBERTa model on Dreamers' 6-class sentiment labels.

The DeBERTa model achieved an overall accuracy of **20.37%**, indicating substantial difficulty in generalizing across fine-grained emotional categories. As visible in the confusion matrix (Figure 9.3), the model overwhelmingly predicted the dominant class (label 6), showing a strong bias toward high-intensity positive sentiments. All other classes—including neutral and moderately negative ones—were completely misclassified.

This behavior can be attributed to two main factors. First, the dataset's limited size and uneven class distribution exacerbate overfitting toward frequent labels. Second, subtle emotional distinctions between neighboring classes (e.g., 4 vs. 5) are often linguistically weak, making it difficult even for advanced language models to detect stable boundaries in short introspective text.

Although LoRA fine-tuning provides efficient adaptation with limited parameters, the results demonstrate that high-granularity sentiment tasks require substantially larger and more balanced datasets to reach reliable discriminative performance.

9.0.6 Cross-Source Evaluation Results: 3-Class Setting

To assess the model's ability to generalize across subjective sentiment annotations, we trained DeBERTa+LoRA classifiers on labels provided by one source and evaluated performance on the other two. This cross-source evaluation enables analysis of label alignment and transferability across annotators with differing psychological perspectives.

The results are summarized in Figure 9.4, where each cell represents the accuracy obtained when training on the source in the corresponding row and testing on the target in the corresponding column. Detailed performance metrics, including precision, recall, and F1-score matrices, are reported in figures 9.5, 9.6 and 9.7 respectively.

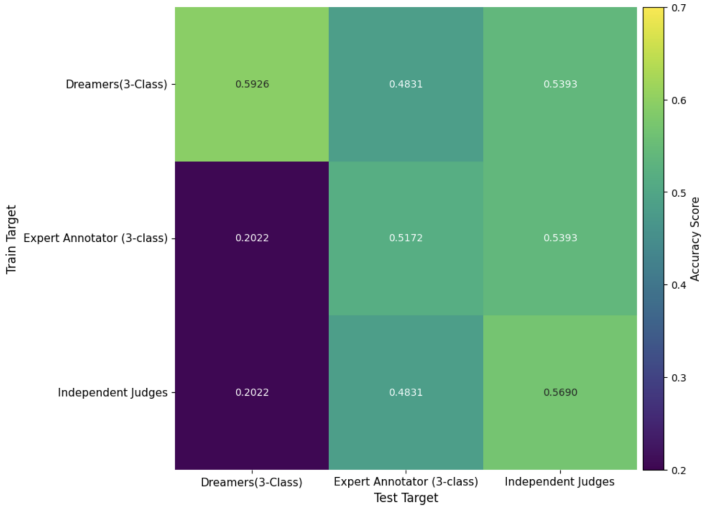


Figure 9.4: Cross-source **accuracy** matrix for 3-class sentiment annotations. Each cell shows the model's accuracy when trained on one rater's labels and tested on another.

Key Findings.

- **Moderate Self-Consistency:** The model trained and evaluated on dreamer self-ratings achieved an accuracy of **59.3%**, indicating moderate internal consistency in introspective sentiment judgments.
- **Asymmetric Transferability from Dreamers:** When trained on dreamers and evaluated on expert or judge annotations, the model reached **48.3%** and **53.9%** accuracy, respectively. This shows a mild improvement over earlier experiments, yet still confirms that emotional representations by dreamers remain only partially transferable to external evaluators.
- **Weak Generalization from External Sources:** Models trained on expert annotations or judges failed to generalize back to the dreamer labels, both achieving only **20.2%** accuracy on the dreamer-labeled test set. This asymmetric pattern highlights a critical mismatch in how emotions are internally experienced versus externally inferred.

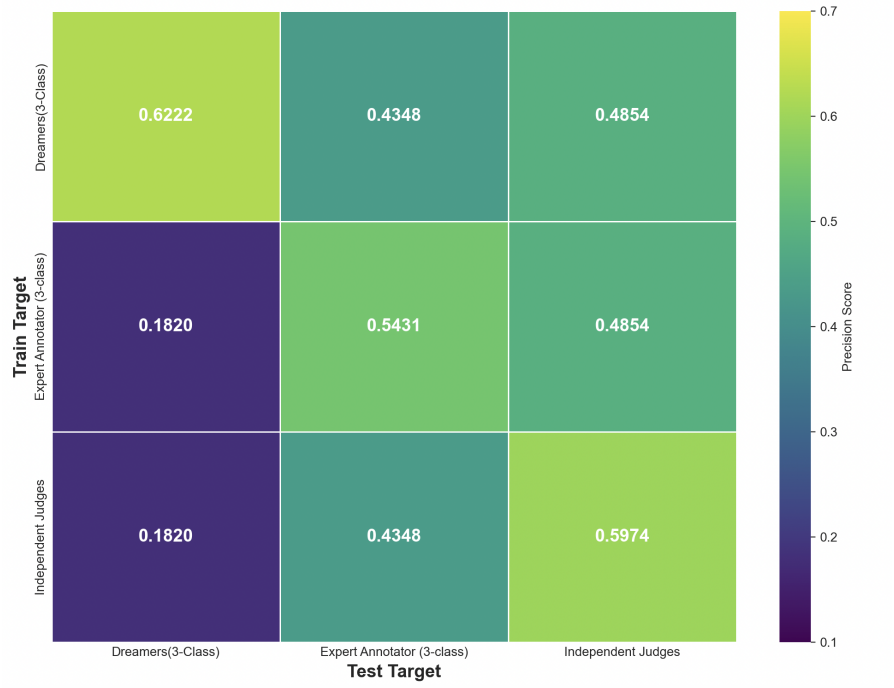


Figure 9.5: 3-Class cross-source **precision** matrix

- **Moderate Cross-Agreement Between Experts and Judges:** Models trained on judges generalized reasonably to expert annotations (**48.3%**) and vice versa (**51.7%**), suggesting a closer alignment between these external raters compared to either's agreement with the dreamers.

These updated results reinforce the central theme of this study: emotional perception is highly dependent on the annotator's perspective. While the expert and judge annotations display a degree of mutual transferability, their ability to model or predict the dreamers' self-rated emotions remains limited. Conversely, although the dreamer-based model transfers only moderately to external raters, it retains some internal coherence.

More importantly, these cross-source generalization patterns are **consistent with our Spearman correlation analysis** (see Sections 8.1.7 and 8.1.9). In particular, we observed a strong rank correlation between expert and judge annotations ($\rho = 0.885$), contrasted with weak correlations between either of them and the dreamers ($\rho = 0.260$ and $\rho = 0.237$). This statistical evidence confirms that self-assessed emotional tone diverges significantly from external evaluation—a central challenge in subjective affective computing tasks such as dream sentiment classification.

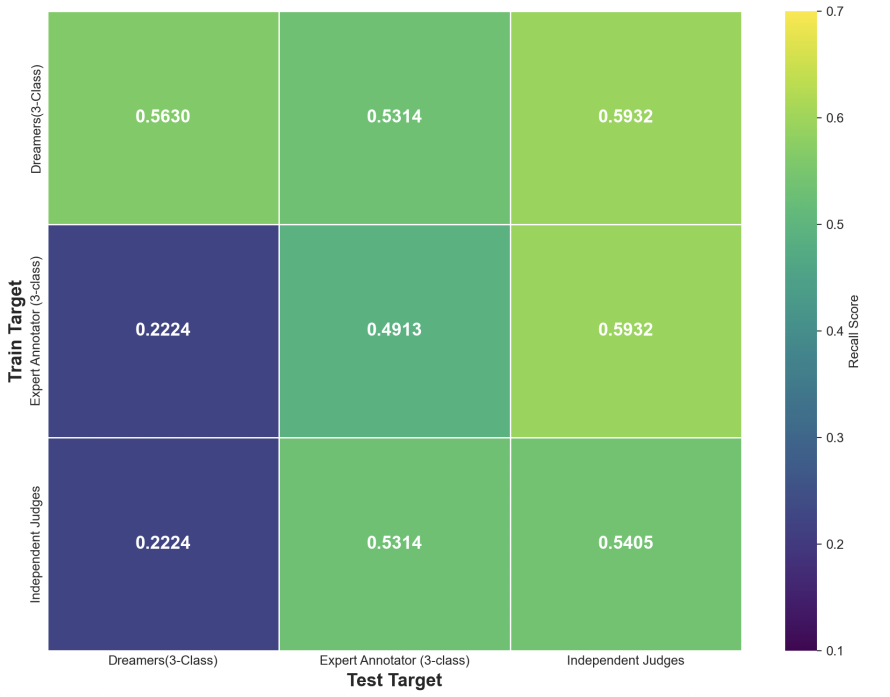


Figure 9.6: 3-Class cross-source **recall** matrix

9.0.7 Cross-Source Evaluation Results: 6-Class Setting

To assess whether finer-grained emotional distinctions could improve cross-rater generalization, we extended our evaluation to the 6-class sentiment annotation scheme. This configuration provides a more nuanced representation of emotional tone but also increases the risk of interpretive variability between annotators.

Figure 9.8 reports the pairwise accuracy scores across dreamer and expert annotations in this setting. Detailed performance metrics, including precision, recall, and F1-score matrices, are reported in figures 9.9, 9.10 and 9.11 respectively.

Key Findings.

- **Overall Low Agreement:** Across all configurations, accuracy scores remain low, with self-consistency values of **20.4%** for both dreamers and the expert annotator. This indicates that even within the same rater, label granularity introduces additional noise and ambiguity, reducing model confidence and consistency.

- **Cross-Rater Generalization:** When trained on dreamers and evaluated on expert labels, the model achieved an accuracy of **20.8%**, while the inverse

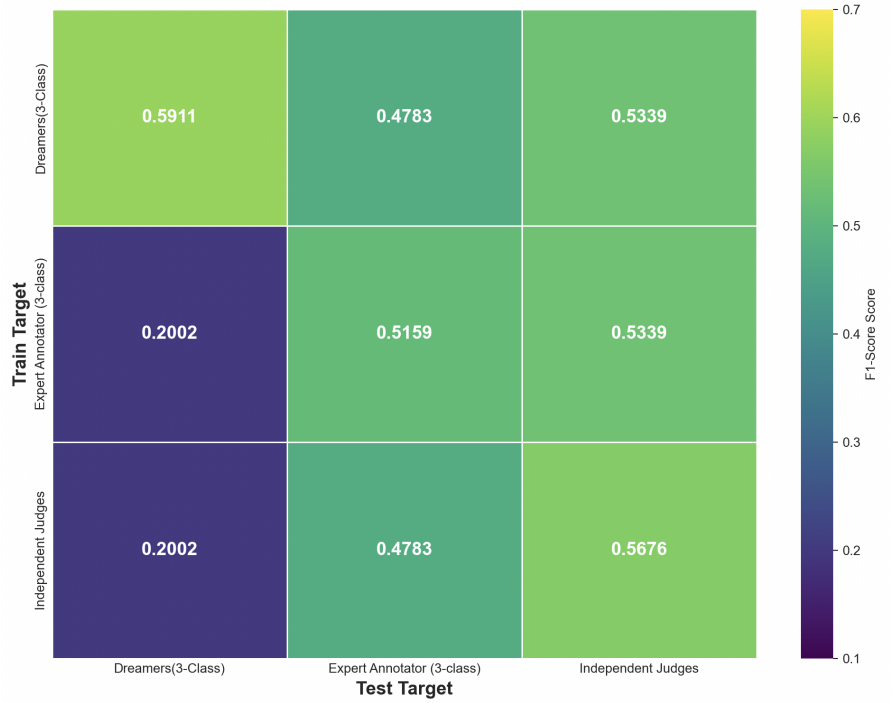


Figure 9.7: 3-Class cross-source **F1-Score** matrix

configuration (expert to dreamer) yielded **18.5%**. Both results reflect a near-random alignment between annotators, suggesting that emotional perception at this granularity is highly subjective and context-dependent.

- **Impact of Fine-Grained Labeling:** Compared to the 3-class setting, performance dropped sharply in all transfer directions, confirming that finer sentiment resolution increases annotation divergence and makes cross-source generalization substantially more difficult.

The 6-class results reveal a pronounced fragmentation in emotional interpretation across annotators. Unlike in the 3-class setup, where partial generalization was observed between external raters, here both self- and cross-source accuracies converge near random levels, emphasizing that fine-grained emotion scales amplify subjective disagreement rather than resolve it.

Consistent with our correlation analysis (see Section 8.1.9), the Spearman rank correlation between Dreamer and Expert labels was extremely weak ($\rho = 0.164$), confirming that these two perspectives exhibit minimal monotonic association. Together, these findings demonstrate that the subjective and context-dependent nature of dream emotions resists stable modeling at high categorical resolution, underlining the intrinsic challenge of affective computing in free-form

psychological narratives.

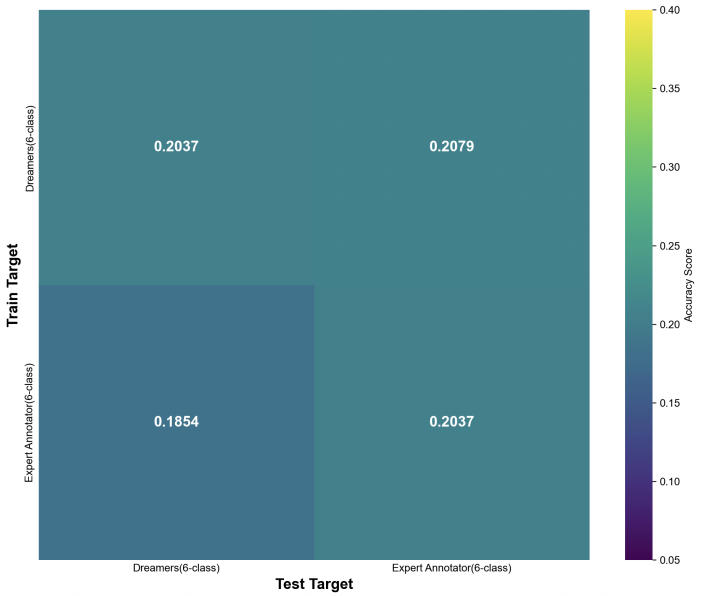


Figure 9.8: Cross-source **accuracy** matrix for 6-class sentiment annotations. Each cell shows the model's accuracy when trained on one rater's labels and tested on another.

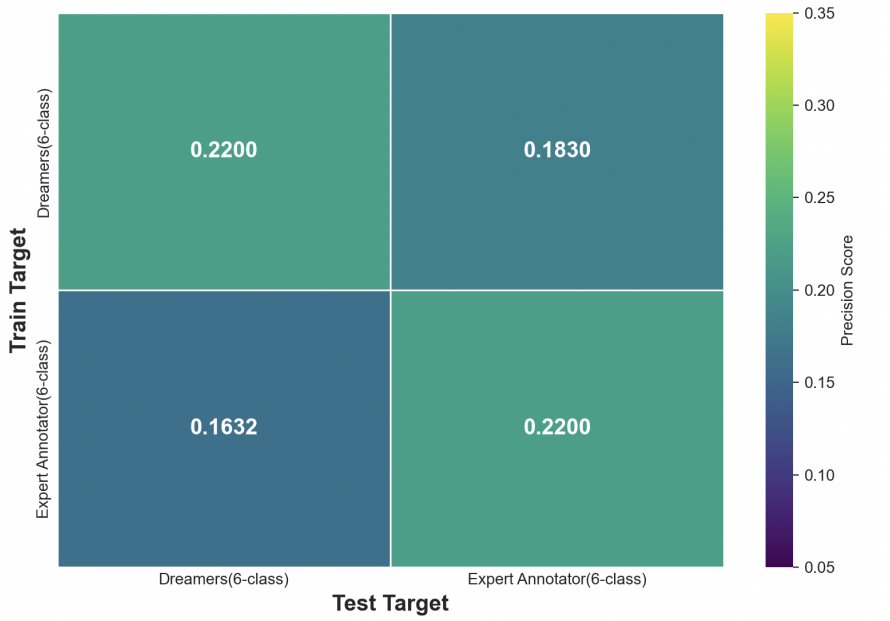


Figure 9.9: 6-Class cross-source **precision** matrix

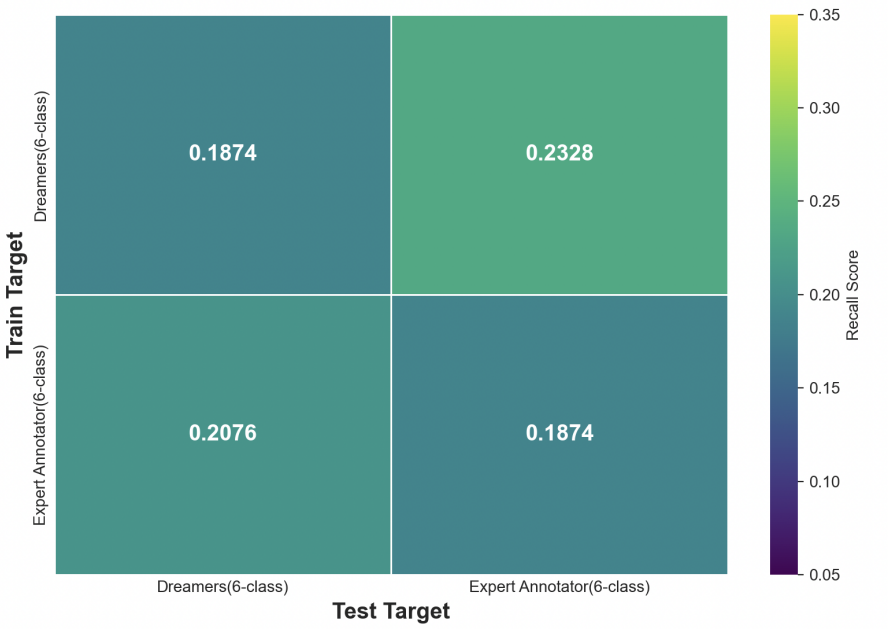


Figure 9.10: 6-Class cross-source **recall** matrix

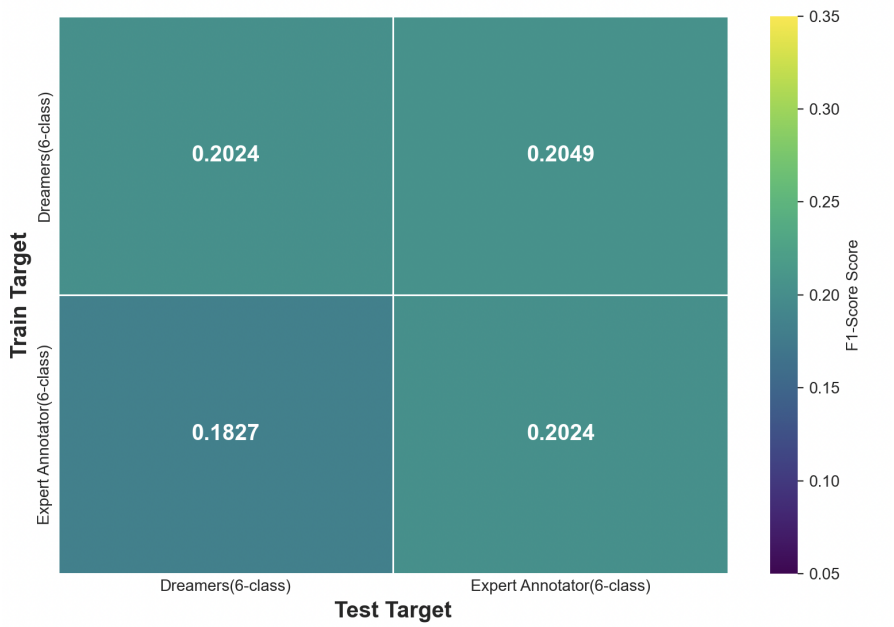


Figure 9.11: 6-Class cross-source **F1-Score** matrix

Chapter 10

Conclusion

This thesis set out to explore whether machine learning can reliably decode the emotional content embedded in free-text dream narratives, a domain marked by abstraction, subjectivity, and linguistic ambiguity. Through a hybrid modeling pipeline combining TF-IDF and Empath features, paired with a carefully tuned Multi-Layer Perceptron and robust resampling strategies, we achieved strong classification performance across both coarse (3-class) and fine-grained (6-class) sentiment scales. The results, 86% accuracy in the 3-class setting and 81% in the 6-class, demonstrate that symbolic and affectively rich dream content is not only learnable but also predictable to a meaningful degree using structured representations of lexical and semantic information.

Beyond performance metrics, this work revealed deeper insights into the human side of emotional annotation. Our cross-source evaluations highlighted a striking lack of agreement between self-reported emotions and those assigned by external annotators. Models trained on one source, such as the dreamers, struggled to generalize to others, with accuracy often dropping below 40%. These findings were reinforced by low Spearman's rank correlations and Cohen's kappa values, which confirmed that divergent interpretations, rather than model limitations, underlie the generalization gaps. In contrast, strong alignment between expert and judge annotations suggests that external perception of emotional tone may be more stable, but also less reflective of the internal experience.

Interestingly, the transformer-based DeBERTa model with LoRA fine-tuning achieved similarly high intra-source performance while maintaining the same cross-source limitations. This convergence of traditional and modern models indicates that regardless of architectural complexity, the ceiling for model generalization is defined not only by data representation, but by the inherent ambiguity and subjectivity of the task.

In essence, this thesis demonstrates that emotional decoding from dreams is computationally feasible but inherently constrained by the nature of human emotion

itself. Sentiment is not a static ground truth; it is negotiated between the dreamer and the observer, shaped by context, psychology, and perspective. Therefore, any AI system operating in this space must be critically aware of these limitations.

Looking forward, future work should focus on multi-annotator fusion strategies, probabilistic labeling schemes, and personalization frameworks that account for subject-specific emotional baselines. Furthermore, expanding the dataset across cultures, languages, and dream types could enrich the generalizability of findings. Ultimately, decoding dreams with AI is not just a computational challenge, it is a philosophical one, requiring us to ask not only what a model predicts, but whose emotion it reflects.

Bibliography

[1] E. Hartmann. «Outline for a theory on the nature and functions of dreaming». In: *Dreaming* 6.2 (1996), pp. 147–170 (cit. on p. 3).

[2] J. A. Hobson, C. D. Smith, and R. Stickgold. «The neuropsychology of REM sleep dreaming». In: *NeuroReport* 9.3 (1998), R1–R14 (cit. on p. 3).

[3] M. Kim, H. Jo, and J. Ryu. «NLP-based Sentiment Analysis of Dream Descriptions and Its Relationship with Psychological Traits». In: *Proc. of the 26th International Conference on Pattern Recognition (ICPR)*. 2022 (cit. on p. 3).

[4] S. Wijeratne, L. Balasuriya, D. Doran, and A. P. Sheth. «Emotion Classification from Short Text: A Comparative Study». In: *Proc. IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. Leipzig, Germany, Aug. 2017 (cit. on p. 3).

[5] D. Nadeau, C. Sabourin, J. De Koninck, S. Matwin, and P. D. Turney. «Automatic Dream Sentiment Analysis». In: *Proc. Canadian Conference on AI (Position Paper)*. Ottawa, Canada, 2006 (cit. on p. 4).

[6] Duyu Tang, Bing Qin, and Ting Liu. «Deep Learning for Sentiment Analysis: Successful Approaches and Future Challenges». In: *WIREs Data Mining and Knowledge Discovery* 5.2 (2015), pp. 292–303 (cit. on p. 4).

[7] Tariq Abdullah and Ahmed Ahmet. «Deep Learning in Sentiment Analysis: Recent Architectures». In: *ACM Computing Surveys* 55.8 (2022), pp. 1–37 (cit. on p. 5).

[8] Ethan Fast, Binbin Chen, and Michael S. Bernstein. «Empath: Understanding Topic Signals in Large-Scale Text». In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM. 2016, pp. 4647–4657. DOI: 10.1145/2858036.2858535 (cit. on p. 12).

[9] Ari Aulia Hakim, Alva Erwin, Kho I Eng, Maulahikmah Galinium, and Wahyu Muliady. «Automated Document Classification for News Article in Bahasa Indonesia Based on Term Frequency Inverse Document Frequency (TF-IDF) Approach». In: *2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE. Yogyakarta, Indonesia, 2014, pp. 1–6. DOI: 10.1109/ICITEED.2014.7003711 (cit. on p. 12).

[10] Lucas B. V. de Amorim, George D. C. Cavalcanti, and Rafael M. O. Cruz. «The Choice of Scaling Technique Matters for Classification Performance». In: *Applied Soft Computing* (2022). Preprint submitted on arXiv:2212.12343. URL: https://arxiv.org/abs/2212.12343 (cit. on p. 13).

[11] Ismael Ramos-Pérez, Álvar Arnaiz-González, Juan J. Rodríguez, and César García-Osorio. «When is resampling beneficial for feature selection with imbalanced wide data?» In: *Expert Systems with Applications* 188 (2022), p. 116015. DOI: 10.1016/j.eswa.2021.116015. URL: https://doi.org/10.1016/j.eswa.2021.116015 (cit. on p. 15).

[12] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. «SMOTE: Synthetic Minority Over-sampling Technique». In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357. URL: https://jair.org/index.php/jair/article/view/10302 (cit. on p. 16).

[13] Marius-Constantin Popescu, Valentina E. Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. «Multilayer Perceptron and Neural Networks». In: *WSEAS Transactions on Circuits and Systems* 8.7 (2009), pp. 579–588. ISSN: 1109-2734 (cit. on p. 20).

[14] Joseph Rynkiewicz. «On Overfitting of Multilayer Perceptrons for Classification». In: *Proceedings of the 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. Bruges, Belgium: i6doc.com, 2019, pp. 257–262. ISBN: 978-2875870650. URL: https://www.i6doc.com/en/ (cit. on p. 21).

[15] Daniel Berrar. «Cross-Validation». In: *Encyclopedia of Bioinformatics and Computational Biology, 2nd edition*. Elsevier, 2024. Chap. 32. DOI: 10.1016/B978-0-323-95502-7.00032-4. URL: https://www.researchgate.net/publication/381674773_Cross-Validation (cit. on pp. 24, 25).

[16] Jerrold H. Zar. «Spearman Rank Correlation». In: *Encyclopedia of Biostatistics*. John Wiley & Sons, 2005. DOI: 10.1002/0470011815.b2a15150. URL: https://www.researchgate.net/publication/227998354_Spearman_Rank_Correlation (cit. on p. 36).

[17] Mary L. McHugh. «Interrater Reliability: The Kappa Statistic». In: *Biochemia Medica* 22.3 (2012), pp. 276–282. DOI: 10.11613/BM.2012.031. URL: https://www.biochemia-medica.com/en/journal/22/3/10.11613/BM.2012.031 (cit. on p. 38).

[18] Pengcheng He, Xiaodong Gao, Jianfeng Chen, and Jingjing Gao. «DeBERTa: Decoding-enhanced BERT with Disentangled Attention». In: *International Conference on Learning Representations (ICLR)*. 2021 (cit. on p. 41).

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.LG]. URL: https://arxiv.org/abs/2106.09685 (cit. on pp. 43, 44).