Politecnico di Torino

Politecnico di Torino

Master of Science in Data Science and Engineering

Academic Year 2024/2025

Graduation Session December 2025

Improving Performance for Multi-Label Classification on Imbalanced Problems using Data Augmentation Techniques

Supervisors:

Flavio Giobergia

Simona Mazzarino

Luca Gilli

Candidate:

Marileni Sinioraki

Abstract

This thesis presents the development of an emotion classification pipeline designed to improve multi-label text classification, with a particular focus on addressing the underrepresentation of minority classes. In multi-label tasks, class imbalance often causes models to perform poorly on infrequent labels compared to majority ones. To mitigate this issue, the proposed approach generates synthetic sentences to enrich minority class samples and enhance overall classification performance. The study begins by establishing a baseline model based on the BERT architecture, trained on the original dataset. A data-driven analysis is then conducted to identify the most representative examples of underperforming labels. These examples are used as input for two data augmentation methods: a traditional synonym replacement technique and a large language model based generation approach. For the latter, different prompting strategies are explored to improve the relevance, quality, and diversity of the generated text. The quality of the synthetic data is evaluated against original samples using appropriate metrics, and the augmented datasets are used to retrain the baseline model to assess performance improvements. The results demonstrate that large language model based augmentation can effectively enhance the performance of minority classes compared to traditional techniques. All code and implementations developed for this work are made publicly available in a GitHub repository to support transparency and reproducibility.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Multi-label text classification is a growing field within Natural Language Processing (NLP), especially in sentiment and emotion recognition, content moderation, and user feedback analysis. In contrast to single-label classification, in which every text is classified into one category, multi-label classification enables multiple labels to be assigned to the same text. This capability shows the complexity of natural language more accurately, but it also makes the problem more complicated.

One of the problems of multi-label text classification is the imbalance of data among different categories. Some labels, such as common emotions or sentiments in the field of sentiment and emotion recognition, like *joy* or *anger*, tend to appear frequently in datasets, while others, such as *disgust* or *anticipation*, may occur far less often. This imbalance can cause models to become biased toward the majority classes, leading to poor performance in predicting rare but potentially critical labels. As a result, the classifier might excel in identifying dominant emotions but struggle to detect subtle or infrequent ones. Addressing data imbalance is critical because minority labels often contain important semantic or contextual information, such as detecting early signs of distress in mental health texts, identifying niche topics in content moderation, or picking up on nuanced opinions in customer feedback.

1.2 Motivation and Objectives

A common approach to deal with data imbalance is *data augmentation*, in which new training examples are generated to enrich the dataset and improve the model's generalization ability. Traditional augmentation techniques, such as synonym replacement, random insertion, word deletion, or back-translation, have been widely used in text-based tasks [1]. However, while these methods can expand the

dataset, they often fail to fully capture the refinement and tone of natural language, sometimes introducing noise or unnatural phrases that reduce model performance. In multi-label emotion classification, this kind of noise can be especially damaging. Even a single corrupted token can change how the emotion of a short sentence is interpreted, creating labels that feel unclear or even contradictory. On top of that, small word-level changes do not alter the broader context or the subtle cues in the text, which are often essential for telling apart emotions that are very similar to each other, such as *annoyance* versus *anger* or *sadness* versus *grief*.

In recent years, large language models (LLMs) has provided a new and more sophisticated way to perform data augmentation. These models can generate contextually rich paraphrases and variations of text that retain the original meaning while introducing linguistic diversity. This allows for more realistic and semantically consistent synthetic data, which can significantly improve the multi-label classifiers. However, the effectiveness of LLM-based augmentation depends on several factors, including the quality of the prompts used to guide the model, the selection of representative sentences, and the evaluation of the generated samples to ensure consistency and relevance. Poorly designed prompts or unfiltered outputs can lead to bias amplification or overfitting in synthetic models. Additionally, LLM-based augmentation tends to be more computationally expensive than traditional approaches. It can also introduce unintended patterns, such as overly polite wording or noticeable shifts in writing style, that were not part of the original dataset. Because of this, it's important to design and evaluate these methods carefully.

The goal of this thesis is to investigate whether data augmentation can improve the performance of multi-label emotion classification models, particularly when working with imbalanced datasets. We compare the impact of traditional augmentation techniques with LLM-based methods, evaluating their effectiveness in improving the classification of minority emotion classes. Furthermore, this work seeks to analyze not only the quantitative effects of augmentation, through metrics such as F1-score, precision, and recall, but also its qualitative impact, evaluating the semantic quality and variability of the generated texts. In doing so, we explicitly distinguish between micro- and macro-averaged metrics in order to understand whether augmentation primarily helps frequent emotions or genuinely benefits rare labels. Finally, the results provide insights into how to optimize data augmentation strategies to address imbalance challenges in NLP tasks, and to what extent generative models are necessary compared to simpler, cheaper baselines.

1.3 Research Problem

This thesis focuses on addressing several key research questions aimed at understanding and mitigating the impact of data imbalance in multi-label emotion

classification tasks:

- How does class imbalance affect the performance of multi-label classification?
- How do traditional augmentation methods compare with LLM-based methods in improving classification performance?
- What role does prompt design play in the quality of synthetic data and its effect on the classifier?
- Can augmentation improve minority-label performance without significantly decreasing performance on majority labels?

To address these questions, the GoEmotions dataset [2] is used. A baseline model based on BERT architecture [3] is initially trained on the entire dataset to establish performance benchmarks. This setting reflects the best achievable performance without any artificial data scarcity and serves as a reference point for all subsequent experiments. To simulate real-world label imbalance scenarios, the dataset is then downsampled, reducing the frequency of certain emotion categories to reflect scarcity conditions while keeping the underlying label correlations intact.

Next, regression analysis is applied to identify labels that perform below expectation relative to their frequency. These labels are then augmented using (i) synonym replacement and (ii) LLM-based paraphrasing with different prompting strategies. This two-stage design makes it possible to check *where* the model struggles (underperforming labels).

Finally, the augmented dataset is used to retrain the baseline model, allowing a comparative evaluation of the impact of each augmentation strategy at the label level and overall performance. By keeping the architecture, optimization procedure, and evaluation pipeline fixed, any observed differences can be attributed primarily to the augmentation strategy. This ensures a systematic analysis of when traditional methods are sufficient, and when LLM-based generation provides clear benefits.

1.4 Structure of the Thesis

The thesis is organized into the following chapters.

`Chapter 2` provides the literature review, covering prior work on multi-label classification and highlighting the main challenges associated with class imbalance. It also discuss existing data augmentation techniques, including both traditional and generative approaches, and presents the GoEmotions dataset and other works that used it.

`Chapter 3` details the methodological framework adopted in this thesis. It explains the preprocessing steps applied to the data, the structure of the experimental

pipeline, and the rationale behind each design decision. The chapter also outlines how the regression analysis is used to identify underperforming labels, and how both traditional and LLM-based augmentation strategies are implemented and integrated into the workflow.

`Chapter 4` presents the experimental setup and results. It discusses the evaluation metrics used to assess model performance, training the BERT base model, the effect of downsampling, and the comparison between traditional and LLM-based augmentation, including different prompting strategies.

`Chapter 5` offers a discussion of the results, highlighting the effectiveness of different augmentation methods and their relative strengths and weaknesses. It also reflects on the importance of prompt design, and shows the limitations observed.

Finally, `Chapter 6` summarize the key insights gained throughout the thesis and outlines directions for future work, suggesting potential extensions such as more advanced augmentation strategies and improved prompt engineering techniques for multi-label data, and applications to other domains.

Chapter 2

Background and Related Works

2.1 Multi-label Text Classification

Multi-label text classification is a task in NLP in which a single text instance can be assigned to multiple labels simultaneously, unlike single-label classification where only one category is predicted [4]. This formulation better shows the complexity of natural language, as texts often have overlapping topics, sentiments, or emotions. Multi-label classification has a big range of applications, including topic categorization, sentiment analysis, and emotion recognition.

Early studies categorized multi-label learning methods into two major groups: *problem transformation* and *algorithm adaptation* approaches [4]. The former converts a multi-label problem into several single-label problems (e.g., Binary Relevance, Classifier Chains), while the latter modifies existing algorithms (such as decision trees or support vector machines) to directly support multi-label outputs. Although these classical approaches set the foundation for multi-label learning, they often struggle with high-dimensional label spaces and fail to capture complex label dependencies [5]. These limitations have caused the shift to methods based on deep learning.

Recent research has used deep neural networks to handle these challenges. Convolutional Neural Networks (CNNs) have been adapted for multi-label text classification by including label clustering and embedding compression to improve scalability on large label sets [5]. Surveys on deep learning for multi-label learning emphasize the growing use of transformer architectures and graph neural networks, which can deal with label correlations and handle sparsity more effectively [6]. In particular, attention mechanisms are currently used a lot. For example, the Label

Attention and Historical Attention model improves discriminative text representations by filtering words using cosine similarity and co-attention mechanisms to capture fine-grained word–label interactions, while historical attention reduces error propagation during training [7]. Similarly, the Label–Sentence Bi-Attention Fusion Network (LSBAFN) [8] combines Bi-LSTM and multi-head attention to extract multi-granularity features (see Figures 2.1 and 2.2).

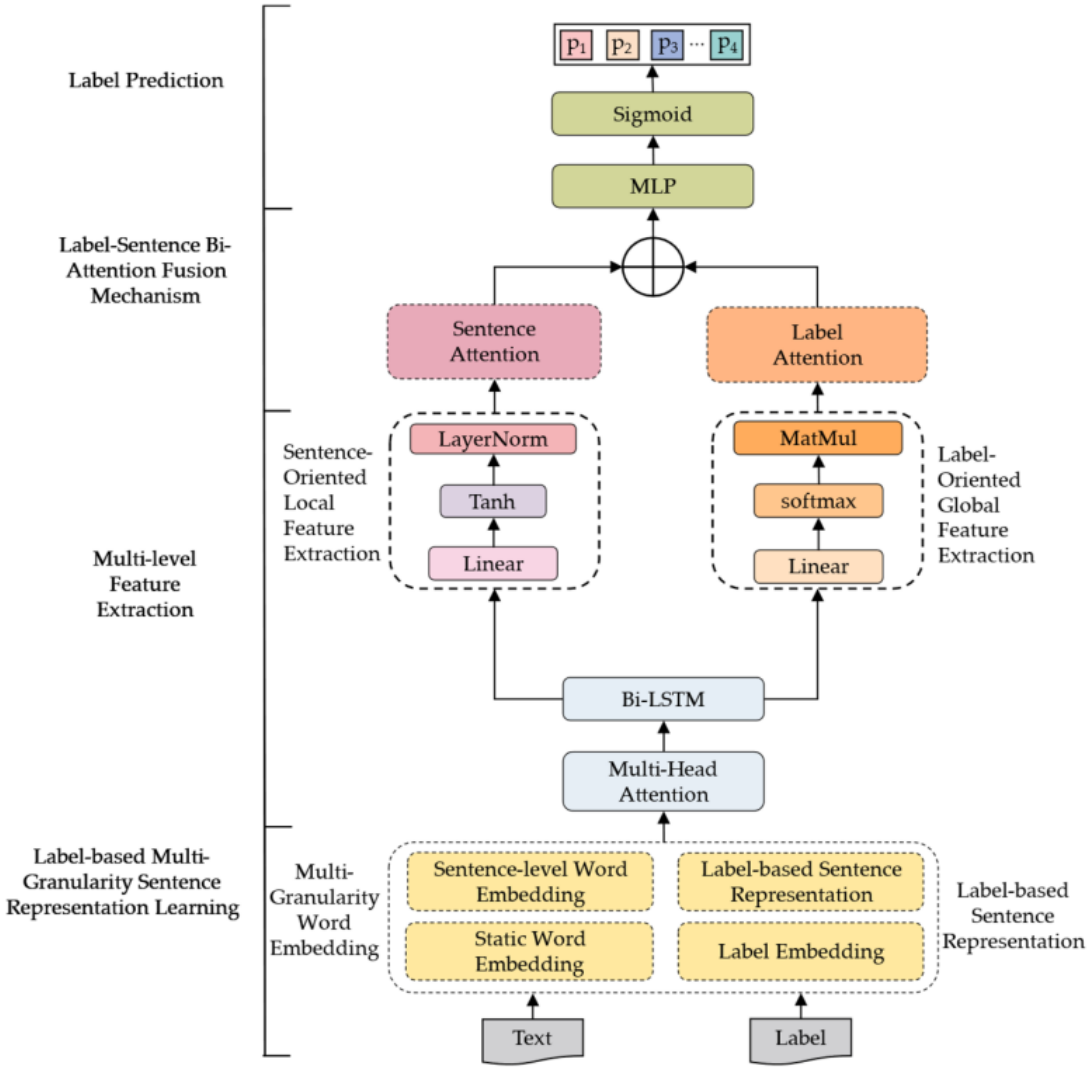

Figure 2.1: Structure of the LSBAFN model (reproduced from Li et al. [8]).

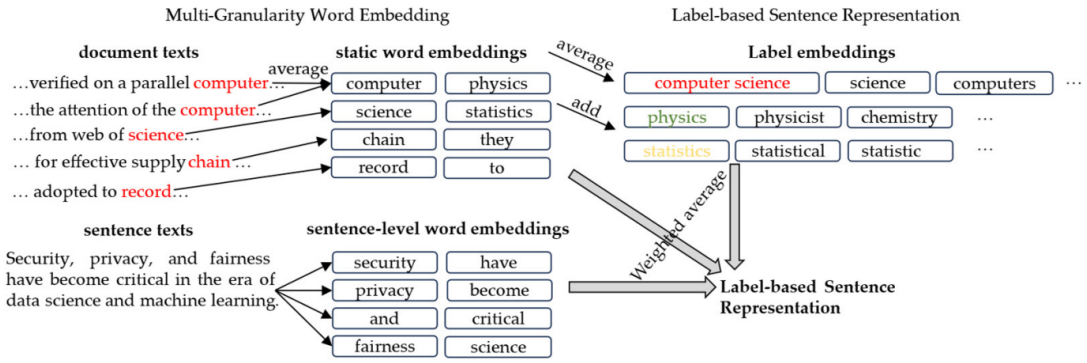

Figure 2.2: Structure of the label-based multi-granularity sentence representation learning (reproduced from Li et al. [8]).

In hierarchical classification settings, multi-label methods can enhance structured label taxonomies to improve accuracy. Surveys have reviewed how models utilize parent–child label relationships to improve predictions across multiple datasets (a list of multiple papers can be found here [9]). Active learning approaches have also been explored. For instance, the Bayesian Expected Confidence-based Active Learning method prioritizes uncertain samples using posterior predictive distributions, reducing annotation costs while maintaining accuracy [10]. Network-based models such as Label Attention and Correlation Networks [7] further address label dependencies through residual blocks and re-weighted binary cross-entropy loss. In few-shot scenarios, where labeled data is scarce, meta-learning and prompt-based methods have been proposed to enable models to adapt efficiently to new labels [11].

Advanced architectures, including 3D attention mechanisms [12], have improved multi-label emotion recognition by jointly modeling spatial and temporal dependencies. In addition, several studies have explored ways to address data imbalance within emotion datasets, such as the use of focal loss or resampling strategies in BERT-based models [3], achieving more balanced performance across both rare and frequent emotions.

Data augmentation plays a complementary role in improving model generalization, particularly in imbalanced scenarios. Surveys have categorized text augmentation techniques, such as paraphrasing, random noising, and back-translation, as effective methods for generating diverse yet semantically consistent samples [13].

Recent work has also integrated augmentation within transformer frameworks, such as Text2Topic [14], a Bi-Encoder Transformer capable of zero-shot multi-label predictions that mitigates imbalance through optimized sampling and label-aware learning (see Figure 2.3).

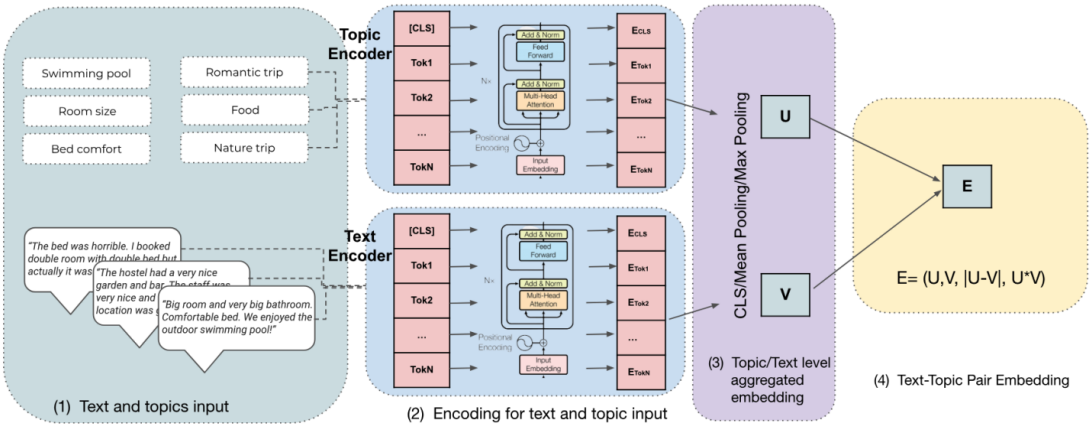

Figure 2.3: Bi-Encoder Concatenation. (1) Input Text-Topic pairs. (2) Encode each text and topic with transformer encoder (the two encoders share weights). (3) Aggregate each of the two text-topic token embeddings into one single vector to represent the topic (U) or the text (V) using CLS/mean-pooling/max-pooling. (4) Combine the two embeddings into one representation of the pair relationship, E. Then feed E into two feedforward layers to get logits output, where BCE loss is applied on (reproduced from Wang et al. [14]).

2.2 Data Imbalance in NLP

Data imbalance is a challenge in NLP, particularly in classification tasks where certain classes appear far less frequently than others [15]. This imbalance often causes models to favor majority classes, resulting in biased predictions and weaker performance for underrepresented categories. The problem is especially notable in real-world datasets, where minority classes frequently correspond to rare events, sentiments, or entities. This can lead to the suffering of the model generalization, and evaluation metrics such as the F1-score tend to decline for minority labels [15].

A survey on deep learning approaches for imbalanced NLP tasks categorizes solutions into three main types: *data-level*, *algorithm-level*, and *hybrid* methods [16]. Data-level techniques focus on modifying the training data through oversampling or undersampling (see Figure 2.4), while algorithm-level approaches adjust the learning process itself using strategies such as cost-sensitive loss functions or threshold moving. Hybrid strategies combine both perspectives to balance data representation and model learning simultaneously. Traditional methods, including random oversampling and text-based adaptations of Synthetic Minority Oversampling Technique (SMOTE) [17], have been explored in several works [18], though these approaches often introduce noise or fail to preserve semantic coherence in linguistic contexts.

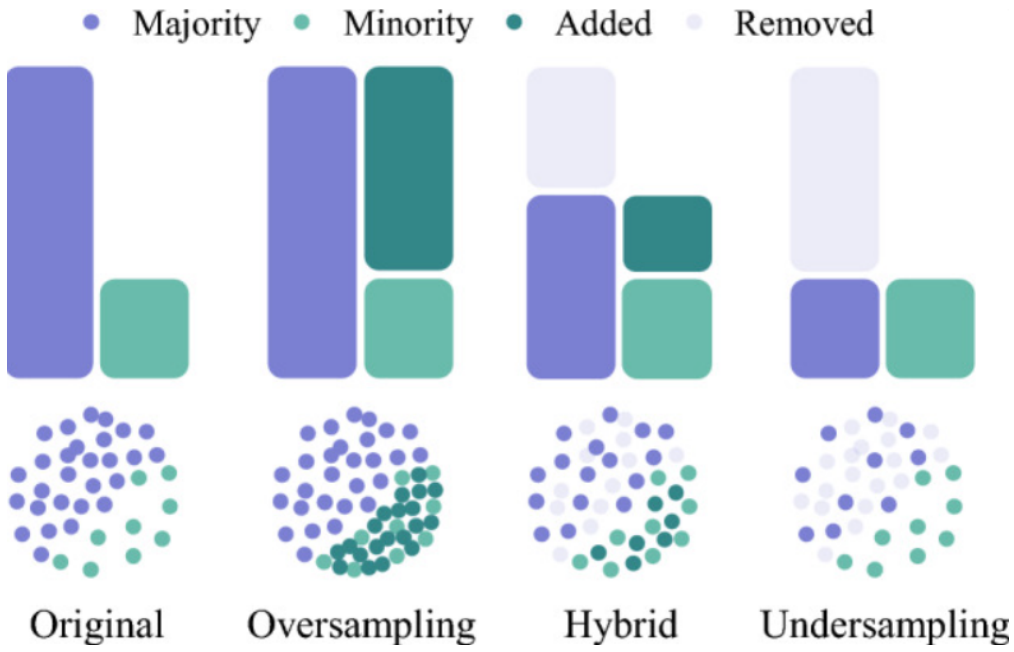

Figure 2.4: Sampling types for imbalanced data preprocessing (reproduced from Werner et al. [16].

Systematic mappings of preprocessing techniques for imbalanced data highlight the effectiveness of synthetic oversampling for improving model accuracy across domains [16]. However, class imbalance remains a difficult issue under out-of-distribution (OOD) conditions, where empirical risk minimization tends to overfit majority classes. To address this, optimization techniques have been proposed to maintain balanced performance under distributional shifts [19]. Recent studies also introduce contrastive sampling strategies that create artificial balance by constructing semantically similar but distinct examples through synonym replacement or sentence mixing, improving performance on text classification tasks [20].

In specialized NLP applications such as Named Entity Recognition (NER), data imbalance has been tackled through selective learning strategies. The Majority-or-Minority (MoM) approach, for example, selectively computes loss functions for specific subsets of data, effectively handling skewed entity distributions without explicit resampling [21]. Oversampling algorithms like adaptive synthetic (ADASYN) [22] have also been evaluated in text-based contexts, showing improvements in recall for rare entities and phrases when used in conjunction with neural architectures (see Figure 2.5) [23].

The progress of LLMs has brought new approaches to addressing imbalance through synthetic data generation. For instance, Moe et al. [24] used LLaMA 3 to generate artificial samples for underrepresented labels in fact-checking datasets,

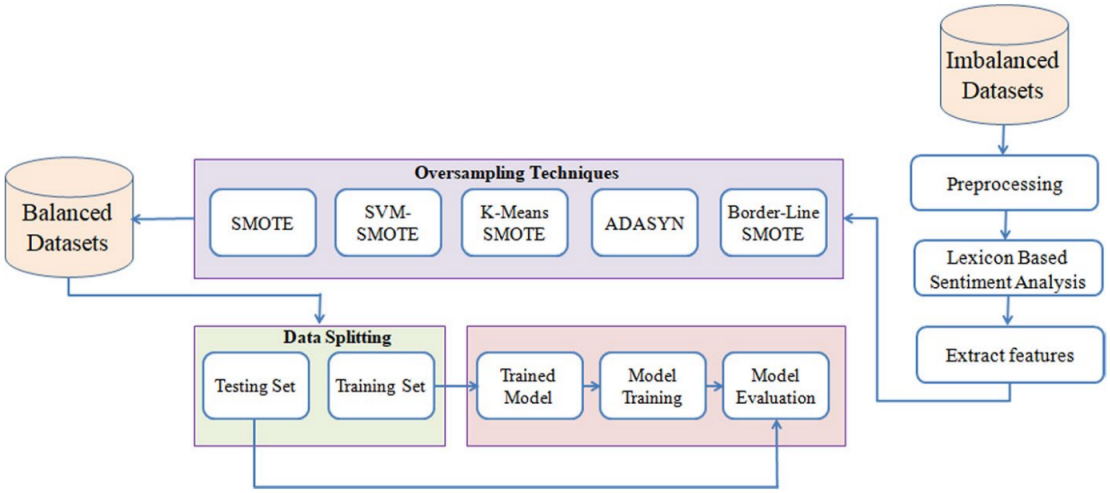

Figure 2.5: Proposed work diagram (ADASYN) (reproduced from Mujahid et al. [23].

resulting in more balanced classification performance. Such generative augmentation techniques demonstrate that LLMs can serve as powerful tools for creating high-quality examples of rare classes, particularly in limited-data settings.

In addition, techniques such as back-translation and Easy Data Augmentation (EDA) have been widely applied in NLP to increase diversity while maintaining semantic trueness [13]. These augmentation strategies are particularly valuable in emotion recognition tasks, where rare emotions must be synthetically enriched to prevent model bias and ensure more balanced performance across all classes.

2.3 Data Augmentation Techniques for Text Data

Data augmentation is used in NLP as a strategy to improve model performance, especially in situations with limited resources or imbalanced datasets. The main idea is to expand the training data by generating new samples that preserve the original meaning while introducing controlled variation. This helps models generalize better and reduces the risk of overfitting [13].

Many techniques fall under the umbrella of data augmentation. Traditional approaches such as SMOTE and ADASYN create synthetic samples to improve the representation of minority classes, while text oriented methods adapt these ideas to linguistic data. Other forms of augmentation rely on modifying the text itself through transformations that maintain semantic consistency. These include paraphrasing, back translation, synonym replacement, and the introduction of small perturbations to the input.

Data augmentation can be achieved through rule based procedures, model driven generation, or a combination of both. Figure 2.6 presents a taxonomy of data augmentation strategies for text, as proposed by [1].

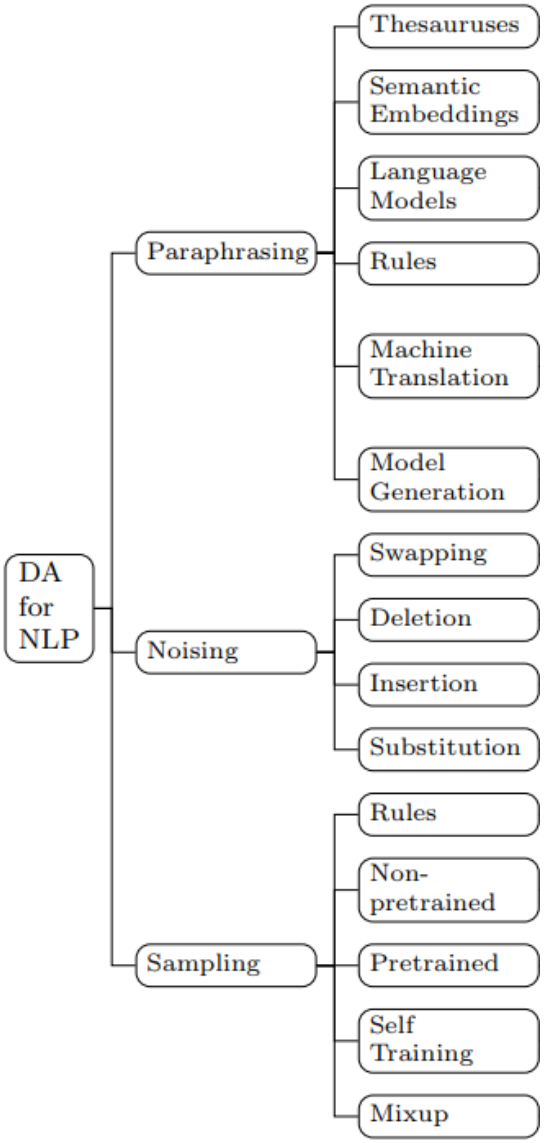

Figure 2.6: Taxonomy of NLP data augmentation methods (reproduced from Li et al. [1]).

Surveys classify data augmentation methods into three main levels: *token-level* (e.g., synonym replacement), *sentence-level* (e.g., sentence shuffling), and *generation-based* (e.g., data produced by generative adversarial networks or large

language models) [13, 25]. Early work such as Easy Data Augmentation (EDA) [26] demonstrated that simple techniques like random insertion, deletion, swapping, and synonym replacement can produce noticeable improvements.

Subsequent research extended these ideas to more specialized NLP tasks. For example, Torres et al. [27] applied augmentation techniques such as mention replacement and contextual embedding swaps to named entity recognition (NER), showing that minority entities could be effectively enriched without introducing artifacts. Similarly, studies in hate speech detection have compared multiple data augmentation strategies, including synonym replacement, contextual embeddings, and back-translation, finding that back-translation (see Figure 2.7) best preserves linguistic nuance while significantly increasing dataset diversity [28].

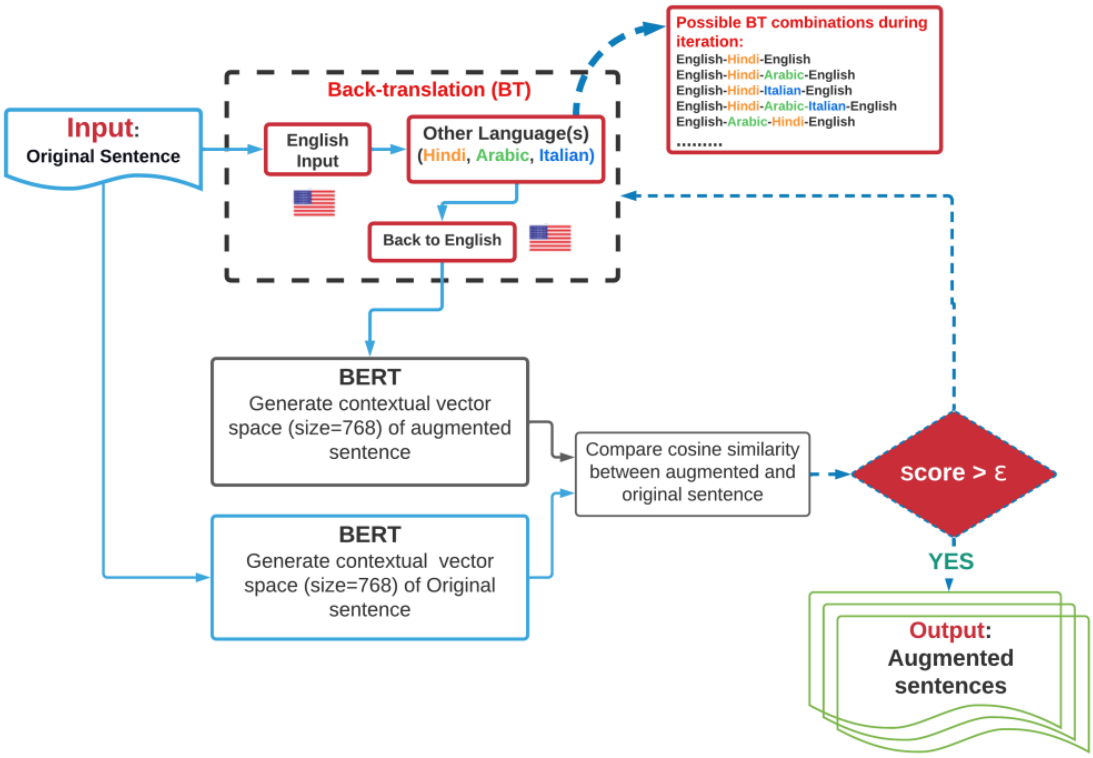

Figure 2.7: Once the new sentence is generated, BERT is used for contextual embedding. Finally, the Cosine similarity is applied to measure the closeness between the original and the augmented sentences (reproduced from Jahan et al. [28]).

The growing availability of large language models has expanded the potential of data augmentation. Surveys on LLM-driven text augmentation categorize current approaches into four main groups: *rule-based* (simple manipulations), *prompt-based*

(zero- or few-shot generation), *model-based* (fine-tuned LLMs for data synthesis), and *hybrid* (see Figure 2.9) [29]. These methods have enabled scalable augmentation pipelines for complex tasks such as multi-label classification. Broader studies across modalities [30] have also highlighted the cross-domain nature of augmentation, drawing parallels between visual and textual transformations, such as rotation in images (see Figure 2.8) and sentence reordering in text.

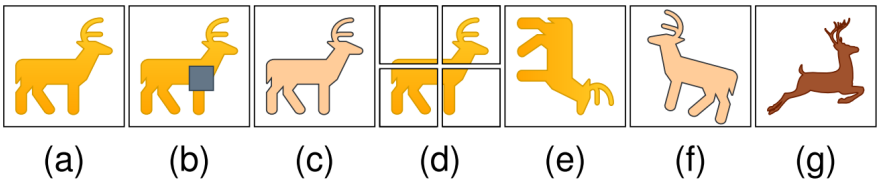

Figure 2.8: A conceptual demonstration of (a) Original Image, (b) Pixel Erasing, (c) Photometric Transformation, (d) Image Cropping, (e) Geometric Transformation, (f) Policy-based Data Augmentation (g) Prompt-based Image Editing (reproduced from Wang et al. [30]).

More advanced frameworks combine augmentation with label correlation modeling. For example, the Text2Topic Bi-Encoder model [14] supports zero-shot multi-label augmentation and fine-grained emotion mapping by integrating sampling optimization with label-aware representations.

Overall, the evolution of data augmentation in NLP, from simple rule-based methods to LLM-based generation, shows its growing significance in addressing class imbalance, improving generalization, and improving performance on fine-grained, multi-label emotion classification tasks.

2.4 Dataset: GoEmotions

The *GoEmotions* dataset [2] is a large-scale, human-annotated corpus developed by Google Research for fine-grained emotion recognition. It contains roughly 58,000 English Reddit comments labeled with 27 emotion categories plus an additional *neutral* label. Because each comment may receive multiple emotion tags, GoEmotions is naturally suited to multi-label classification. A central challenge of the dataset is its long-tail label distribution: a small number of emotions (including *neutral* and other high-frequency categories) account for a large share of annotations, while emotions such as *grief*, *disgust*, and *pride* are rare. This skew can cause emotion classifiers trained naively on the full dataset to favor dominant classes and underperform on minority emotions.

Demszky et al. [2] introduced GoEmotions and established strong multi-label baselines. Their main baseline fine-tuned a BERT model for multi-label prediction,

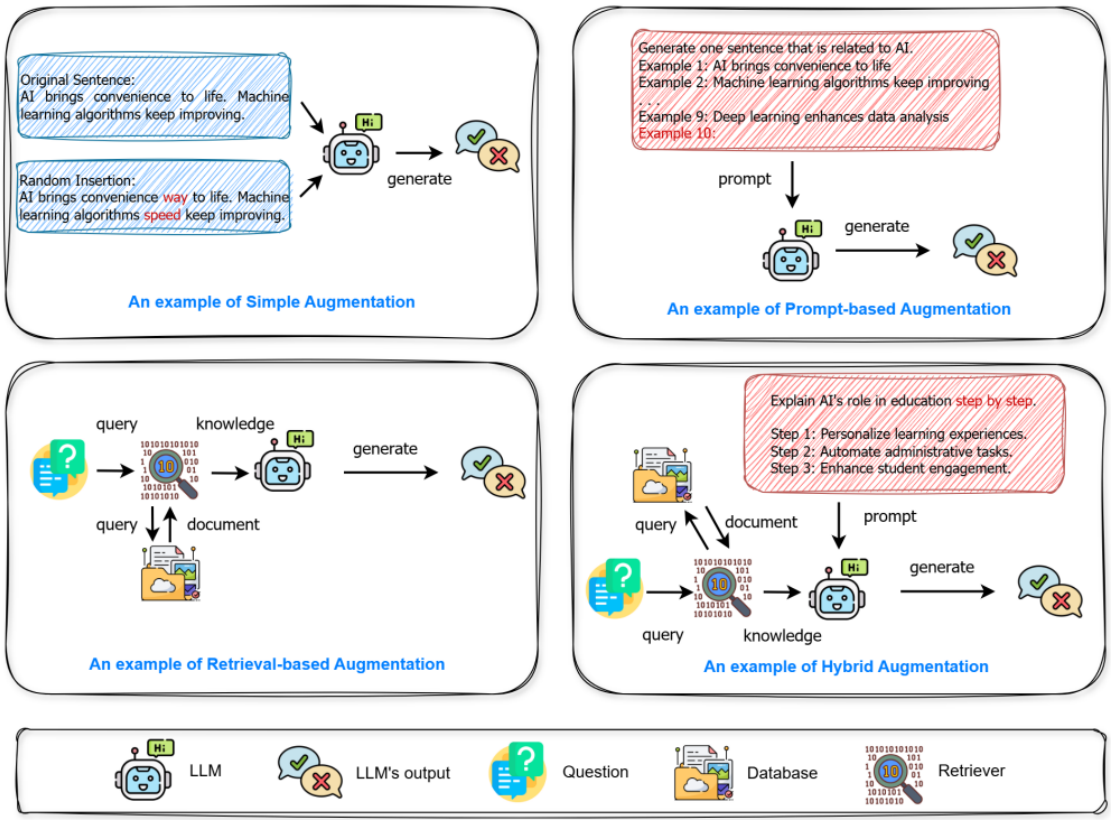

Figure 2.9: Four categories of data augmentation techniques (reproduced from Chai et al. [29]).

achieving a macro-F1 of about 0.46 on the full 28-label taxonomy, while a biLSTM baseline performed notably worse (around 0.41 macro-F1) [2]. The authors also emphasized the pronounced imbalance across labels and documented that several low-frequency emotions receive near-zero F1 under standard training, motivating the need for imbalance-aware learning and evaluation.

Subsequent work has aimed to improve performance on GoEmotions through model-level and training-level changes. Bashynska et al. [31] trained a BERT-encoder classifier for GoEmotions and reported an improved macro-F1 (around 0.51), confirming the benefits of transformer-based representations, but their results still reflect weaker performance on rare classes under the original label skew. Luo et al. [12] proposed a label-aware 3D attention mechanism (3-CA) that models interactions across emotion-specific attention planes; on GoEmotions, their approach raises macro-F1 to roughly 0.56 compared to standard transformer baselines [12]. Focusing explicitly on imbalance, Ramakrishnan and Babu [32] introduced a clipped asymmetric loss on top of BERT to down-weight easy majority-label predictions

and emphasize minority-label errors, yielding a clear macro-F1 gain (about 0.54 overall) and more reliable detection of rare emotions [32].

Alongside these modeling advances, data augmentation has emerged as a complementary way to mitigate GoEmotions' long-tail distribution. Wang et al. [33] explored large language model (LLM) based synthetic data generation combined with transfer learning for GoEmotions, showing that augmenting training data can improve fine-grained multi-label emotion recognition beyond non-augmented baselines. Ahanin et al. [34] systematically compared classical EDA-style methods [26], BERT-based contextual substitution, and ChatGPT-based augmentation on GoEmotions; they found that contextual BERT augmentation produced the most consistent benefit, improving macro-F1 by about 5 to 6% in their setup [34]. More broadly, comparative model studies on GoEmotions (e.g., stacked LSTM and transformer variants) confirm that even strong architectures remain sensitive to label skew and that rare emotions remain the primary source of error [35].

Overall, GoEmotions remains a challenging but productive benchmark for advancing multi-label emotion classification. Prior work has either emphasized imbalance-aware objectives and architectures [32, 12] or explored augmentation as a separate route to robustness [33, 34]. Surveys on class imbalance in NLP and text augmentation similarly treat these as related but often independently addressed problems [15, 13, 1]. This gap motivates the present thesis, which investigates how targeted data augmentation can be integrated with regression-based label performance analysis to more directly improve minority-class recognition within a balanced, multi-label framework.

For the experiments conducted in this thesis, the GoEmotions dataset was downloaded directly from the official release URLs provided by the original authors [2]. It was integrated into the project's preprocessing pipeline and split into training, validation, and test sets following the original partitioning for comparability with prior benchmarks. Standard preprocessing, including text normalization and BERT tokenization [3, 36], was applied before training. These standardized splits and preprocessing steps provide a foundation for the baseline, downsampling, and targeted data augmentation experiments presented in later chapters.

Chapter 3

Methodology

3.1 Preprocessing

This section describes how the original GoEmotions dataset was transformed into a clean, reproducible multi-label dataset. It explains each stage of the process, from getting the data and verifying its integrity, to cleaning, consolidating emotion labels, exploring the data, and finally obtaining the final dataset. All source code developed for this thesis, including data preprocessing, model training, and augmentation techniques, is provided in an open-source repository on GitHub (`https://github.com/Clearbox-AI/Marileni_Sinioraki_Thesis/tree/main`).

3.1.1 Dataset Acquisition

The preprocessing begins by downloading the official GoEmotions dataset , which is provided in three separate CSV files. These files are combined into a single dataset so that all annotated comments can be processed in one place. During the merge, the original comment IDs are kept intact to ensure that each entry can be uniquely identified later, even though the overall index is updated to ensure consistency. This results in a clean, unified dataset that is ready for further analysis.

Before any preprocessing or filtering, the raw dataset includes the following columns:

- `text`: the Reddit comment text,
- `id`: a unique identifier for each comment,
- `author`: the Reddit username of the comment author,
- `subreddit`: the community where the comment was posted,

- `link_id` and `parent_id`: identifiers linking the comment to its thread and parent post,
- `created_utc`: the timestamp of comment creation,
- `rater_id`: the annotator identifier for the emotion labels,
- `example_very_unclear`: a quality flag indicating comments with ambiguous or uncertain emotional content,
- 27 binary emotion indicator columns representing each emotion category (*admiration*, *amusement*, *anger*, *annoyance*, *approval*, *caring*, *confusion*, *curiosity*, *desire*, *disappointment*, *disapproval*, *disgust*, *embarrassment*, *excitement*, *fear*, *gratitude*, *grief*, *joy*, *love*, *nervousness*, *optimism*, *pride*, *realization*, *relief*, *remorse*, *sadness*, *surprise*),
- one additional binary column for the `neutral` label.

Each row in the original dataset represents one person's annotation of a Reddit comment where annotators were all native English speakers from India. As a result, the same comment can appear more than once if multiple raters labeled it. Because of this, it is necessary to combine and consolidate those labels during preprocessing.

To illustrate the original data structure, Table 3.1 shows a small sample of the original GoEmotions CSV file before preprocessing. Each row corresponds to a Reddit comment annotated with one or more emotion indicators. The columns include id, text content, and binary emotion labels.

text	**id**	**sadness**	**love**	**remorse**
That game hurt.	eew5j0j	1	0	0
The ABC's hard-hitting investigation :/ Such a sad article.	ed2mah1	1	0	0
Man I love reddit.	eeibobj	0	1	0
Pity. I had some decent lunches there, but never went there at night.	ee04wu6	0	0	1
I am so sorry for your loss and all the stress you have right now. All the internet hugs you want right now! <3	ee04wu6	0	1	1

Table 3.1: Excerpt from the raw GoEmotions dataset (simplified view).

3.1.2 Cleaning and filtering

Two filtering steps were applied to improve reliability of the label and better align the dataset with the objectives of emotion recognition:

1. Rows where the column `example_very_unclear = True` are discarded to eliminate instances that annotators marked as confusing or unreliable.

2. All rows labeled as `neutral` are removed to keep the focus on explicit emotions and to emphasize emotion expressions rather than emotionally neutral content.

These filters are applied *before* label unification (explained more in detail in the following section) to ensure that uncertain or neutral examples do not influence the final multi-label representations. In the following Table 3.2, we can take a look at the remaining rows after each filtering step.

Filtering step	**Remaining rows**
Original concatenated dataset	211,225
Remove unclear examples	207,814
Remove neutral examples	152,516

Table 3.2: Dataset size after each preprocessing step.

Per-`id` aggregation

As mentioned before, each Reddit comment in the GoEmotions dataset may appear multiple times because it was annotated independently by several raters. To obtain a single entry per comment, the dataset is grouped by the unique identifier `id`. This ensures that all annotations referring to the same comment are merged into one record.

For each comment, the first occurrence of the `text`, `subreddit`, and `created_utc` values is retained, since these fields are identical across raters. For each emotion label $e \in \mathcal{E}$, where $\mathcal{E}$ is the set of 27 emotions (excluding `neutral`), the binary annotations from different raters are first aggregated by computing their mean:

$$\overline{r}_{i,e} = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{i,e}^{(j)},$$

where $y_{i,e}^{(j)}$ is the label assigned by rater j for emotion e on comment i, and n_i is the number of raters for that comment.

A final binary label is then assigned according to a unanimity rule:

$$y_{i,e} = \mathbb{1}(\overline{r}_{i,e} = 1.0)\,,$$

meaning that an emotion e is considered present for comment i only if *all* raters selected it. If even one rater did not mark the emotion, it is excluded ($y_{i,e} = 0$). In this way, we obtain reliable, high-confidence emotion labels.

The resulting dataset contains one row per unique comment `id`, with the structure:

$$\{\texttt{id}, \texttt{text}, \texttt{subreddit}, \texttt{created\_utc}\} \cup \{y_e\}_{e \in \mathcal{E}},$$

where each $y_e \in \{0,1\}$ represents the final binary presence or absence of emotion e under unanimous agreement.

This aggregation criterion minimizes label noise and ensures that only comments with clear emotion are included in the dataset. No further text normalization (such as lowercasing or stopword removal) is performed at this stage, as these operations are being done to the tokenization step.

After applying this aggregation rule and removing comments with no active emotion labels, the final dataset contains **53,740** unique comments. Table 3.3 and Figure 3.1 shows the number of samples associated with each of the 27 emotion labels in the resulting multi-label dataset. As shown, certain emotions such as *approval*, *annoyance*, and *admiration* are relatively frequent, while others like *relief* and *grief* remain underrepresented, reflecting the class imbalance of the GoEmotions dataset.

3.2 Pipeline Overview

The proposed method is implemented in a pipeline designed to process the raw GoEmotions dataset, build a clean multi-label emotion corpus, train and evaluate a baseline model, analyze label performance under simulated data scarcity, and finally assess the impact of different data augmentation strategies. The entire workflow follows a reproducible structure, allowing each stage to be executed either independently or as part of a complete automated run.

In the code, the pipeline is organized into different components, each implemented in a specific module: `prepare_data.py` (data acquisition and preprocessing), `data_loader.py` (data formatting and tokenization), `model.py` (definition of the classification model), `trainer.py` (training and evaluation routines), and `utils.py` (auxiliary functions and configuration utilities). The main entry script manages the execution flow and contains command line arguments that allow users to control each step of the process.

At a high level, the pipeline contains the following phases:

Table 3.3: Label distribution in the final aggregated GoEmotions dataset (27 emotions, 53,740 comments).

Emotion	Count
approval	13,253
annoyance	10,038
admiration	9,937
disapproval	8,402
realization	7,250
disappointment	6,661
curiosity	6,216
optimism	6,209
joy	5,718
anger	5,652
gratitude	5,364
confusion	5,321
amusement	5,187
sadness	4,676
love	4,383
excitement	4,355
caring	4,339
disgust	4,058
surprise	3,830
desire	2,838
fear	2,142
embarrassment	2,004
remorse	1,664
nervousness	1,557
pride	1,128
relief	1,085
grief	560

1. Downloading, cleaning, and aggregating the original GoEmotions release into a unified multi-label dataset.
2. Transforming textual data into a ready input using the BERT tokenizer [3], with a train, validation and test split stratified by subreddit to reduce topic bias.
3. Fine-tuning the BERT base model on the preprocessed dataset to establish reference performance.

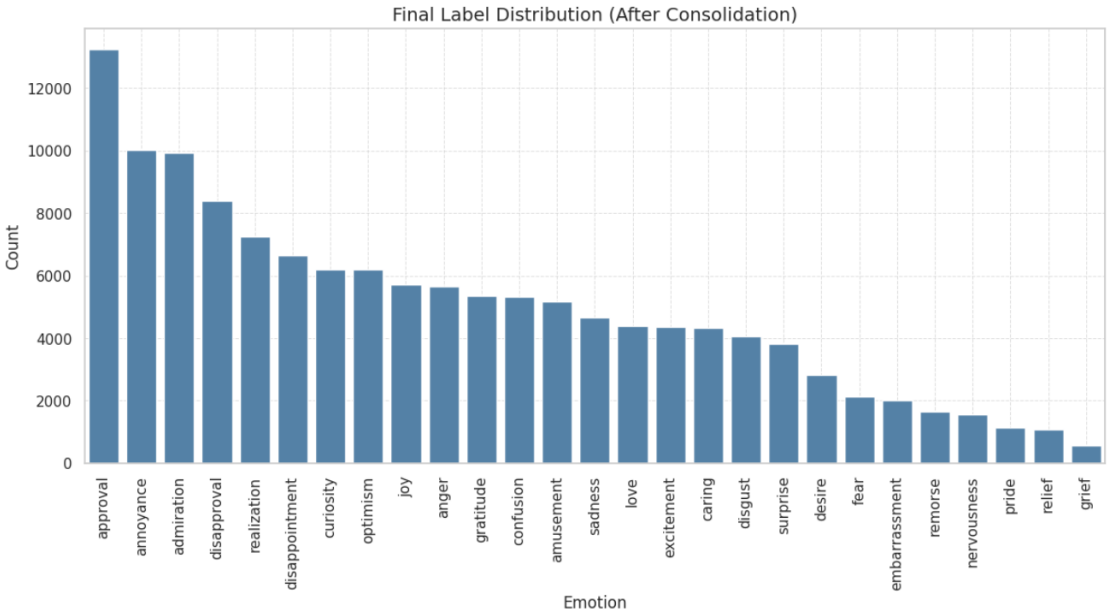

Figure 3.1: Labels distribution.

4. Optimizing classification thresholds on the validation set to better balance precision and recall across multiple emotion labels.

5. Assessing the model on the test set and comparing results against default decision thresholds.

6. Reducing the number of samples per label to simulate class imbalance and analyze how data scarcity affects model performance.

7. Applying a regression-based approach to relate label frequency and F1-score, identifying which emotions are most affected by data scarcity.

8. Generating additional samples for underperforming labels using both traditional synonym replacement (Easy Data Augmentation, EDA) and LLM–based rewriting (Mistral-7B Instruct [37]), followed by retraining and evaluation to measure performance gains.

This structured approach (see Figure 3.2) ensures that each experiment can be reproduced, extended, or modified. This design also allows future integration of new models, datasets, or augmentation techniques.

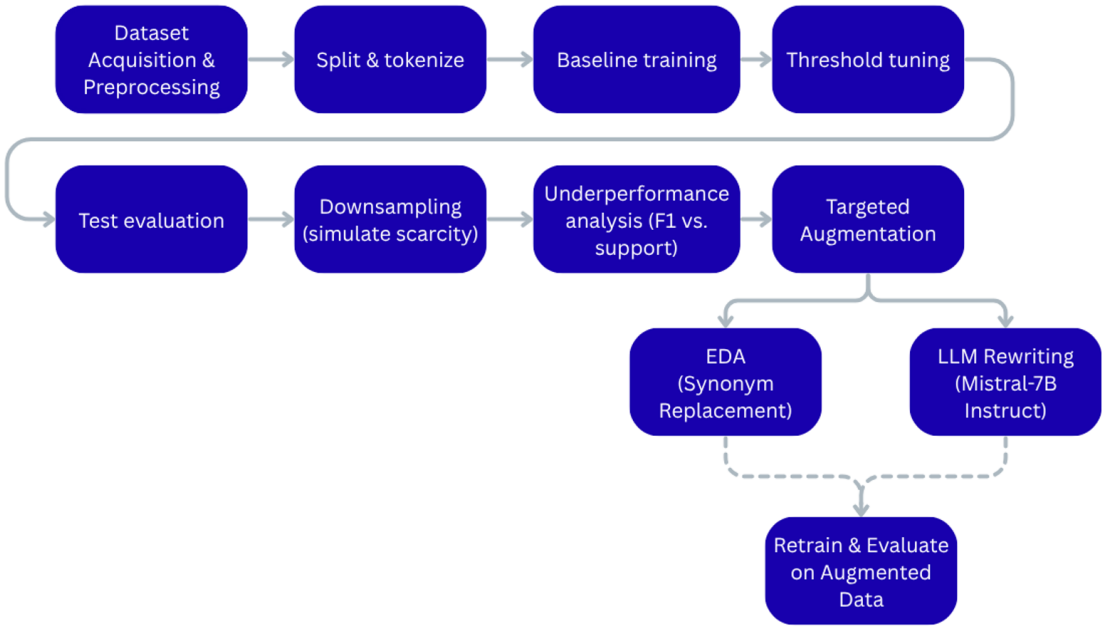

Figure 3.2: Pipeline from data preparation to baseline training, threshold tuning, and evaluation; followed by downsampling, regression-based diagnosis of underperforming labels, targeted augmentation (EDA and LLM), and retraining on the enriched dataset.

3.2.1 Loading, splitting, and tokenization

At this stage, the processed dataset is being prepared for model training. To create the train, validation, and test sets, the data is **split within each subreddit**. This means that every subreddit contributes examples to all three splits in proportions defined by the specified ratios. This approach ensures that each split preserves the diversity of topics and styles present across subreddits.

The final dataset includes comments from **483 unique subreddits**, covering a wide range of themes and emotional tones. Some communities are very active, contributing hundreds of comments, while others appear only a few times. Table 3.4 and Figure 3.3 provides an overview of the subreddit distribution. The most represented communities, such as *loveafterlockup* and *cringe*, each contribute over 200 examples, while smaller subreddits like *farcry* contain only a handful of posts.

Once the dataset is analyzed, it is split into training, validation, and test sets following a 60/20/20 ratio. The final splits contain **31,967** examples for training, **10,844** for validation, and **10,929** for testing. Each subset is then converted into a Hugging Face `Dataset` object, which stores both the text and a `labels` field of a 27-dimensional binary vector representing the presence or absence of each emotion.

Before model training, each comment is transformed from raw text into the input

Table 3.4: Subreddit distribution in the final GoEmotions dataset (top 10).

Subreddit	**Number of posts**
loveafterlockup	227
cringe	226
socialanxiety	224
AnimalsBeingBros	218
confessions	212
vanderpumprules	211
danganronpa	206
90dayfianceuncensored	205
90DayFiance	203
datingoverthirty	200
Unique subreddits	483
Minimum posts per subreddit	22
Number of subreddits with 22 posts	1
Subreddit with fewest posts	farcry (22)

format required by BERT using the Hugging Face `AutoTokenizer`, configured for the `bert-base-uncased` model. `Bert-base-uncased` is one of the most widely used transformer-based architectures introduced by Devlin et al. [3], which itself is based on the transformer encoder proposed by Vaswani et al. [38]. BERT is designed to capture rich contextual representations of language by jointly conditioning on both left and right contexts within a sentence. This bidirectionality enables the model to understand subtle dependencies between words, which is important for emotion classification, where meaning often depends on context.

The `base` configuration of BERT consists of 12 transformer encoder layers, each with 12 self-attention heads and 768-dimensional hidden representations, resulting in approximately 110 million parameters. The model was pre-trained on two large-scale corpora, the BooksCorpus and English Wikipedia, using a masked language modeling (MLM) and next sentence prediction (NSP) objective [3].

Additionally, BERT employs a subword tokenization method known as *WordPiece*, originally introduced by Google for neural machine translation [39]. This approach decomposes words into smaller units from a fixed-size vocabulary of 30,522 tokens. It enables the model to effectively represent both common and rare words without excessively increasing vocabulary size. For instance, the rare word *overgeneralization* may be segmented into `over`, `##general`, and `##ization`, where the prefix `##` marks continuation subwords.

The `uncased` variant of BERT applies automatic lowercasing and accent removal,

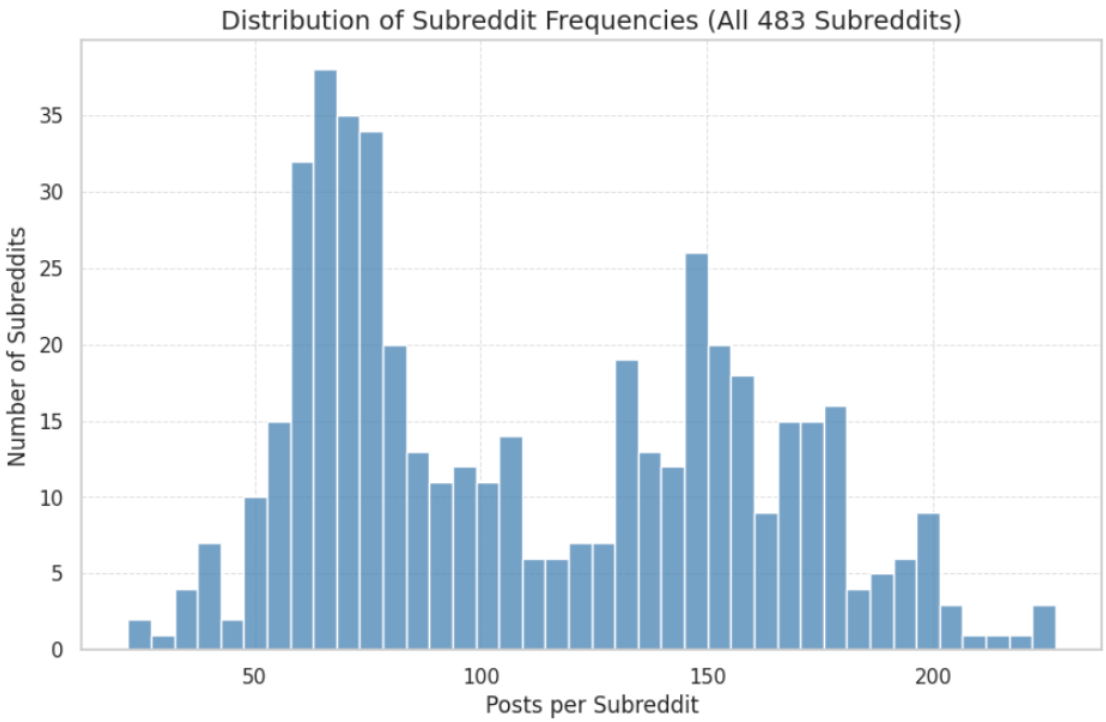

Figure 3.3: Subreddits distribution.

mapping variations such as *Happy*, *HAPPY*, and *happy* to the same token sequence. This reduces sparsity and it is the standard for English classification tasks where capitalization does not carry strong semantic meaning. During tokenization, BERT also inserts two special tokens: the classification token `[CLS]` at the beginning of each sequence and the separator token `[SEP]` at the end. Each input sequence therefore takes the following general form:

$$\texttt{[CLS]}\ t_1\ t_2\ \ldots\ t_n\ \texttt{[SEP]},$$

where t_i denotes the individual WordPiece tokens. During fine-tuning, the hidden representation associated with the `[CLS]` token serves as the aggregated sequence embedding used by the classification head.

To ensure uniform input lengths across all samples, both **padding** and **truncation** are applied. Each sequence is capped at a maximum of 128 tokens. Longer texts are truncated from the end, while shorter ones are padded using the special `[PAD]` token until they reach the fixed length. Alongside this, the tokenizer generates an *attention mask* (a binary vector of the same length) where 1 indicates valid tokens and 0 marks padded positions. The mask prevents BERT's self-attention mechanism from attending to padded tokens. The choice of 128 tokens represents a balance between covering most Reddit comments and keeping memory and training

time manageable. In practice, the majority of GoEmotions samples are shorter than this limit.

After tokenization, all examples are stored in the Hugging Face `Dataset` format, which integrates with PyTorch. Each processed example contains the following fields:

- `input_ids`: the token indices corresponding to subword tokens,
- `attention_mask`: a binary mask distinguishing valid tokens from padding,
- `labels`: a 27-dimensional binary vector representing the emotion categories.

Then it is passed directly to the Hugging Face `Trainer`, which handles data batching, GPU allocation, and gradient accumulation. Overall, this step transforms the dataset from structured text into model-ready tensors.

3.2.2 Baseline training

In this thesis, the BERT model is fine-tuned for multi-label emotion classification. Since each Reddit comment may express multiple emotions simultaneously, the output layer is modified to include 27 independent sigmoid units, one for each emotion label. Unlike a softmax layer, which enforces mutual exclusivity among classes, the sigmoid activation allows each label to be activated independently. The model outputs a probability vector $\mathbf{p} = [p_1, p_2, \ldots, p_{27}]$, where $p_i \in [0,1]$ represents the predicted probability of emotion i being present in the text. The loss function used is the binary cross-entropy with logits loss (`BCEWithLogitsLoss`), defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\Big[y_{ij}\log(\sigma(\hat{y}_{ij})) + (1-y_{ij})\log(1-\sigma(\hat{y}_{ij}))\Big],$$

where N is the number of samples, K is the number of emotion labels, y_{ij} is the ground truth label for class j in example i, $\hat{y}_{ij}$ is the model's raw output (logit), and $\sigma(\cdot)$ is the sigmoid activation function. This formulation treats each emotion prediction as an independent binary decision.

Training is conducted using the Hugging Face `Trainer` API [36], which simplifies the fine-tuning process by handling gradient accumulation, learning rate scheduling, evaluation, and checkpointing. The model is trained with a 27-unit sigmoid head by setting `problem_type=multi_label_classification` in the `BertForSequenceClassification` class. The training loop is implemented through a lightweight wrapper around the `Trainer`, replacing the default loss with `BCEWithLogitsLoss`.

Training parameters such as batch size, learning rate, weight decay, number of epochs, and warmup ratio are defined using the `TrainingArguments` class.

Validation is performed at the end of each epoch to monitor progress and prevent overfitting.

The baseline model has two purposes. First, it provides a reference performance on the original dataset without any data augmentation, establishing a benchmark for later comparison. Second, it forms the foundation for the downsampling experiments, where the dataset is reduced to simulate data scarcity. Retraining the baseline under these conditions helps quantify how performance deteriorates when minority labels are underrepresented. As expected, the reduction in training data affects rare emotions, which shows the importance of synthetic data generation for balancing the label distribution.

Table 3.5: Hyperparameters and Training Arguments for Fine-Tuning BERT.

Model	**Parameter**	**Value**
BERT-base-uncased	Learning rate	2×10^{-5}
	Optimizer	AdamW
	Adam (β_1, β_2)	(0.9, 0.999)
	Adam ϵ	1×10^{-8}
	Weight decay	0.01
	Warmup ratio	0.1
	LR scheduler	linear
	Epochs	3
	Train batch size	16
	Eval batch size	32
	Max sequence length	128
	Gradient accumulation	1
	FP16 training	False
	Save strategy	epoch
	Logging strategy	epoch

The hyperparameter configuration used for fine-tuning BERT (see table 3.5) follows recommendations from the original BERT paper [3], from optimisation studies such as AdamW [40], and from best practices in the Hugging Face Transformers ecosystem [36]. The learning rate is set to 2×10^{-5}, a value widely recognised as a stable choice for BERT fine-tuning. Larger learning rates can overwrite pretrained knowledge and lead to unstable convergence, while smaller ones often slow down learning or cause underfitting. The AdamW optimizer is used because it decouples weight decay from the gradient update rule, improving generalisation and addressing limitations of the original Adam algorithm [41]. The default Adam parameters $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$ are retained, as these settings have consistently proven effective for transformer architectures and modifying them rarely yields

benefits.

A weight decay of 0.01 is applied to reduce overfitting, which is particularly important when the fine-tuning dataset is much smaller than the corpora used during pretraining. Training begins with a warmup phase covering the first 10% of optimization steps, gradually increasing the learning rate to stabilise the early stages of fine-tuning. After warmup, a linear learning rate decay schedule is employed, which gently decreases the learning rate toward zero and is known to encourage smooth convergence for transformer models.

The model is trained for three epochs, consistent with the recommendation in [3] that BERT typically converges within 2–4 epochs for sentence-level classification tasks. The training batch size is set to 16, a common choice that balances gradient stability with GPU memory limitations, while evaluation uses a larger batch size of 32 to speed up inference. The maximum sequence length is restricted to 128 tokens, as most Reddit comments in the GoEmotions dataset fall well below this threshold; shorter sequences improve computational efficiency and reduce unnecessary padding.

Gradient accumulation is not used, as memory constraints allow training with a batch size of 16 directly. Mixed precision (FP16) training is disabled to avoid potential numerical instability, ensuring reproducibility and consistency across machines. Both saving and logging are performed once per epoch, which avoids excessive checkpointing, keeps logs interpretable, and aligns naturally with the validation and threshold-tuning procedures used later in the pipeline.

Overall, this hyperparameter configuration reflects a standard, robust, and empirically validated setup for BERT fine-tuning. Using such well-established settings ensures that the baseline serves as a fair reference point, enabling improvements to be attributed to the data augmentation techniques rather than to hyperparameter optimisation.

Threshold tuning on the validation set

In multi-label classification, each label is predicted independently, and a threshold must be applied to convert probabilities into binary outputs. Using a fixed threshold of 0.5 for all labels often leads to suboptimal results, especially under class imbalance. To address this, a per-label threshold tuning procedure is performed on the validation set. For each emotion, the precision and recall is computed, and the threshold that maximizes the F1 score is selected (the metrics are explained in the next chapter in more detail). These optimal thresholds replace the default 0.5 values for all subsequent evaluations, ensuring that each emotion is classified according to its own empirical decision boundary. This tuning step improves both macro- and micro-F1 scores by aligning decision thresholds.

Test evaluation and default-threshold comparison

After threshold optimization, the model is evaluated on the test split using the tuned thresholds. Performance is reported using micro- and macro-averaged precision, recall, and F1 scores, providing perspectives on model performance across common and rare labels. The test set is also evaluated using the default threshold of 0.5 to quantify the benefit of threshold tuning. Additionally, a per-class evaluation report is produced, summarizing precision, recall, F1, and support for each of the 27 emotions. These metrics reveal which emotions are consistently well captured and which remain challenging due to limited representation or label ambiguity.

3.2.3 Downsampling experiments (simulated scarcity)

To evaluate the model under conditions of limited training data, a series of downsampling experiments were conducted to simulate label scarcity. In these experiments, the training set was progressively reduced by a user-specified percentage per emotion label, ranging from 10% to 90% of the original samples. Downsampling was applied approximately uniformly across labels, ensuring that each emotion retained the same proportion of its original examples. This approach preserves the multi-label structure of the dataset, meaning that comments with multiple emotions continue to reflect realistic overlaps, even after reduction.

For each downsampling level, the experimental pipeline repeats the entire process from scratch: tokenization, model initialization, fine-tuning, validation threshold tuning, and evaluation on the fixed test set. The resulting metrics are aggregated into tables and plotted as F1-score versus data reduction curves (check the next chapter). This setup is used for doing a study of how class frequency affects both global and per-label performance, particularly for minority emotions that are already underrepresented.

Among the various reduction levels tested, the configuration corresponding to a 60% reduction (i.e., retaining 40% of each label) was selected for detailed analysis. This choice represents a realistic scenario: the dataset remains large enough for stable model training while clearly exhibiting the effects of scarcity on minority emotions. At this reduction level, we have comparisons with data augmentation experiments performed later in this study.

Table 3.6 summarizes the resulting label distribution after applying the 60% downsampling. Each entry reports the number of samples per emotion before and after reduction, along with the percentage of data retained.

3.2.4 Underperforming-label analysis via regression

After training the baseline model under the specific data reduction setting (60% downsampling), the next step is to identify which emotion labels are most negatively

Table 3.6: Label distribution after 60% downsampling (40% remaining per label).

Emotion	Original Count	New Count
approval	7915	3166
annoyance	6006	2402
admiration	5879	2351
disapproval	4982	1992
realization	4350	1740
disappointment	3928	1571
optimism	3743	1497
curiosity	3710	1484
joy	3395	1358
anger	3382	1352
gratitude	3185	1274
confusion	3149	1259
amusement	3063	1225
sadness	2764	1105
love	2640	1056
caring	2594	1037
excitement	2566	1026
disgust	2380	952
surprise	2311	924
desire	1694	677
fear	1244	497
embarrassment	1199	479
remorse	973	389
nervousness	944	377
pride	687	274
relief	646	258
grief	340	136

affected by limited data. To do this, a regression-based analysis is performed that models the relationship between the number of samples available for each emotion (*label support*) and the corresponding per-class F1 score obtained on the test set. A simple linear regression is fitted to predict the expected F1 score from the logarithm of label frequency, following prior work showing that model performance in imbalanced classification tasks typically scales with label frequency in a roughly logarithmic fashion [42].

Labels whose observed F1 scores fall significantly below the regression line are considered *underperforming relative to their support*. In simpler terms, these

are emotion classes that don't perform as well as someone would expect given how much data they have. This usually happens because of subtle differences in meaning, emotions that overlap with others, or a lack of variety in the training examples. Spotting these specific labels helps focus improvements where they are really needed, instead of just applying data augmentation to every emotion equally.

3.2.5 Augmentation experiments (EDA and LLM-based generation)

Once the most underperforming labels are identified, two complementary augmentation strategies are applied to improve their representation in the downsampled dataset. In this way there is a check whether the new augmented data can reduce the effects of class imbalance and improve overall model performance. Both approaches are designed to increase diversity and quantity in low-resource labels while preserving the underlying emotional semantics.

The first method, *traditional data augmentation (EDA – Easy Data Augmentation)*, follows the approach introduced by Wei and Zou [26]. Specifically, the synonym replacement technique substitutes a small proportion of non-critical words in a sentence with WordNet [43] based synonyms, generating label-consistent paraphrases that maintain the same emotional tone and the same meaning. Some examples for the emotion embarrassment can be shown in table 3.7. This lightweight augmentation technique has been shown to be effective in text classification tasks by increasing linguistic variety without requiring large computational resources [13, 44]. In the context of this thesis, EDA is applied only to the emotion categories identified as underperforming, and the number of new examples is scaled until each targeted label reaches the predefined support of the most frequent emotion.

The second method uses a *large language model* for the generation of the synthetic data in order to augment the baseline dataset. Synthetic data refers to information that is artificially created to resemble real observations while avoiding a direct copy of the original material. It maintains the statistical properties and semantic patterns of the real dataset, making it suitable for training machine learning models in situations where certain categories are rare or difficult to collect. Recent work has highlighted the usefulness of synthetic text for improving performance in low resource and imbalanced scenarios [45].

Specifically, the `Mistral-7B-Instruct` model is used to rewrite real examples into semantically similar sentences conditioned on the target emotion. Some examples for the emotion embarrassment can be shown in table 3.8. The LLM is prompted with an instruction enabling it to generate diverse, fluent, and emotion-preserving paraphrases. Previous studies have shown that instruction-tuned LLMs can effectively produce high-quality labeled data for NLP tasks, including emotion classification and sentiment analysis [24, 33]. Unlike traditional rule-based EDA,

Original text	Augmentated text
Looks kinda creepy when you keep watching	Looks kinda spooky when you keep watching
He may be embarrassed about the condition of their house.	He may be ashamed about the condition of their house.
His interviews are so awkward	His interviews are so clumsy
Sorry I forgot a word! I added it back in, thanks for letting me know	Sorry I forgot a term! I added it back in, thanks for letting me know
Y'all been eatin' long enough now, stop being greedy!	Y'all been eatin' long enough now, stop being selfish!

Table 3.7: Excerpt from the traditional data augmentation technique.

this approach uses deep contextual understanding to generate richer, more natural examples that better capture the ways people actually express themselves.

Also in this case, synthetic examples are added incrementally until the target label reaches the support of the most frequent emotion. The augmented training dataset is then used to train the BERT base model under the same settings as the baseline. Threshold tuning is performed again on the unchanged validation set, and evaluation is conducted on the same fixed test split to ensure a fair comparison across models. The pipeline records key performance metrics, including micro- and macro-averaged precision, recall, and F1, as well as the number of synthetic samples generated for each label.
The hyperparameters used for Mistral-7B-Instruct (see table 3.9) were selected to balance three main goals: (i) preserving the original emotion of each sentence, (ii) generating linguistically natural rewrites suitable for social media text, and (iii) ensuring computational feasibility during large-scale augmentation.

The model `mistralai/Mistral-7B-Instruct-v0.2` is an instruction-tuned, decoder-only language model with 7 billion parameters. Instruction tuning improves controllability and consistency under prompting, making it more reliable for tasks such as emotion-conditioned paraphrasing. The model is loaded in `torch.float16` to reduce memory consumption while maintaining high generation quality, following common practice in LLM inference pipelines.

The setting `device_map="auto"` automatically distributes model components across available GPUs. This avoids manual device placement and ensures efficient use of hardware resources, which is especially helpful for 7B-parameter models.

A generation batch size of 8 prompts was chosen as a compromise between throughput and GPU memory constraints. Allowing up to 40 new tokens ensures

Original text	Augmentated text
Looks kinda creepy when you keep watching	Feels a bit awkward to keep staring at that.
He may be embarrassed about the condition of their house.	He might feel embarrassed about the state of their house.
His interviews are so awkward	The interviews make him come across as uncomfortable.
Sorry I forgot a word! I added it back in, thanks for letting me know	Apologies for the oversight, I've added the missing word back in. Embarrassed for the mistake.
Y'all been eatin' long enough now, stop being greedy!	It's time to show some restraint, y'all have been eating enough.

Table 3.8: Excerpt from the LLM-based augmentation technique.

that rewritten sentences remain close in length to the originals while avoiding overly long or drifting generations.

Nucleus sampling with `top-p = 0.95` and temperature 0.7 was selected to encourage lexical diversity without sacrificing semantic fidelity. Lower temperatures produced overly literal rewrites, while higher temperatures increased the risk of semantic drift.

A chat-style prompt structure combining a system message with a user message was used to stabilize model behaviour. This format aligns with Mistral's instruction-tuning setup and has been shown to improve faithfulness and adherence to constraints in rewriting tasks. The system prompt explicitly restricts stylistic changes and prohibits the introduction of new content, encouraging emotion-preserving variations rather than uncontrolled paraphrases.

Each rewrite is conditioned on both the original sentence and the target emotion. Only the target emotion is activated in the synthetic label vector, ensuring clean supervision when the synthetic examples are merged with the training set. The number of generations per label is defined by the `target_count` parameter, which aligns augmentation with the imbalance level identified in earlier analysis.

Short or malformed generations (fewer than three tokens) are discarded to ensure that only meaningful rewrites contribute to the training set. This post-processing step is necessary because LLMs occasionally produce incomplete outputs, particularly when conditioned on very short input sentences.

Together, these choices aim to produce high-quality paraphrases that preserve emotional content while increasing label diversity. This configuration achieved substantially better results than simpler synonym-based augmentation methods

Table 3.9: Generation configuration for Mistral-7B-Instruct used in LLM-based data augmentation.

Setting	**Value**
Model	`mistralai/Mistral-7B-Instruct-v0.2`
Model type	Decoder-only causal LM (instruction-tuned)
Parameters	7B
Precision	`torch.float16`
Device mapping	`device_map="auto"`
Task	Sentence rewriting for emotion-preserving augmentation
Batch size (generation)	8 prompts per batch
Max new tokens	40
Sampling strategy	Nucleus sampling (`do_sample=True`)
Temperature	0.7
Top-p	0.95
Pad token	EOS token (`pad_token_id = llm_model.config.eos_token_id`)
Random seed	42 (Python and Torch)
Prompt format	System + user messages (chat template)
System prompt	Instruction to rewrite social media sentences while preserving emotion
Conditioning signal	Original sentence + target emotion label
Per-label target count	`target_count` (desired support per label)
Output filtering	Discard generations with fewer than 3 tokens
Synthetic label vector	Single active emotion: $y_e = 1$ iff $e =$ label

and earlier prompt versions, showing the importance of careful hyperparameter and prompt design in LLM-based data generation.

Comparison The traditional EDA approach (Table 3.7) produces minimal lexical changes that preserve sentence structure and overall semantics. This method tends to substitute individual words with synonyms (e.g., 'creepy' → 'spooky', 'embarrassed' → 'ashamed'), resulting in relatively conservative augmentations. While this keeps the augmented data close to the original, it may introduce limited linguistic diversity and stylistic variation.

In contrast, the LLM-based augmentation technique (Table 3.8) generates paraphrases that differ more substantially from the original text. LLMs often restructure sentences, introduce new expressions, and adapt tone, leading to more natural and contextually rich variations. For example, "Looks kinda creepy when

you keep watching" becomes "Feels a bit awkward to keep staring at that," which is a full paraphrase rather than a synonym substitution. This allows for greater diversity in training data, which can improve model strength, though it also increases the risk of semantic drift if not monitored.

Overall, EDA provides controlled and predictable lexical variation, while LLM-based augmentation offers broader linguistic flexibility and more human-like paraphrasing.

Original Text	**EDA Augmentation**	**LLM-Based Augmentation**
Looks kinda creepy when you keep watching	Looks kinda spooky when you keep watching	Feels a bit awkward to keep staring at that.
He may be embarrassed about the condition of their house.	He may be ashamed about the condition of their house.	He might feel embarrassed about the state of their house.
His interviews are so awkward	His interviews are so clumsy	The interviews make him come across as uncomfortable.
Sorry I forgot a word! I added it back in, thanks for letting me know	Sorry I forgot a term! I added it back in, thanks for letting me know	Apologies for the oversight, I've added the missing word back in. Embarrassed for the mistake.
Y'all been eatin' long enough now, stop being greedy!	Y'all been eatin' long enough now, stop being selfish!	It's time to show some restraint, y'all have been eating enough.

Table 3.10: Comparison between EDA-based and LLM-based augmentation strategies.

This design allows for a direct comparison among three experimental conditions: (1) the downsampled baseline (no augmentation), (2) the dataset augmented with EDA-based synonym replacements, and (3) the dataset augmented with LLM-generated rewrites. Comparing these scenarios gives us information on the effectiveness of traditional versus modern augmentation approaches in enhancing minority and overall label performance.

3.2.6 Reproducibility

All stages can be invoked separately through command-line switches: `-download`, `-prepare`, `-analyze`, `-train`, `-downsample`, `-analyze-labels`, and `-augment-experiment`. Model and data hyperparameters (model-name, epochs, batch-size, learning-rate, downsampling levels, augmentation target counts, and augmentation type) are exposed as flags to support ablations. The pipeline creates required directories on demand, caches dataset downloads, seeds random generators where applicable, and writes deterministic outputs (processed CSV, plots, per-experiment results), making runs repeatable across environments.

Chapter 4

Experiments and Results

4.1 Evaluation Metrics

To evaluate the performance of the model, several standard metrics were used in multi-label classification: precision, recall, and F1-score. Since these metrics can be averaged across labels in different ways, we considered both *micro* and *macro* averaging. Using both, we come up with a more complete view of how the model behaves, particularly when dealing with imbalanced classes.

Precision measures the proportion of predicted positive instances that are actually correct. For a single label i, precision is defined as:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}$$

where TP_i denotes the number of true positives for label i, and FP_i denotes the number of false positives.

Recall measures the proportion of actual positive instances that are correctly identified by the model. For a single label i, recall is defined as:

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i}$$

where FN_i denotes the number of false negatives for label i.

F1-score is the harmonic mean of precision and recall, providing a balanced measure that accounts for both false positives and false negatives. For a single label i, it is given by:

$$\text{F1}_i = \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

In **micro-averaging**, the contributions of all classes are aggregated to compute the overall metric. This approach gives equal weight to each individual prediction,

making it sensitive to class imbalance and dominated by majority classes. Micro-averaged metrics are defined as:

$$\text{Precision}_{micro} = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)}$$

$$\text{Recall}_{micro} = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)}$$

$$\text{F1}_{micro} = \frac{2 \cdot \text{Precision}_{micro} \cdot \text{Recall}_{micro}}{\text{Precision}_{micro} + \text{Recall}_{micro}}$$

In **macro-averaging**, the metric is computed independently for each label and then averaged across all labels. This gives equal weight to each class, regardless of its frequency, making it particularly useful for evaluating performance on minority classes. Macro-averaged metrics are defined as:

$$\text{Precision}_{macro} = \frac{1}{L} \sum_{i=1}^{L} \text{Precision}_i$$

$$\text{Recall}_{macro} = \frac{1}{L} \sum_{i=1}^{L} \text{Recall}_i$$

$$\text{F1}_{macro} = \frac{1}{L} \sum_{i=1}^{L} \text{F1}_i$$

where L is the total number of labels.

Using both micro and macro averaging gives a more complete picture of the model's performance. Micro-averaged metrics show how well the model works overall, but they are mostly driven by the labels that appear most often. Macro-averaged metrics, instead, emphasize how the model performs on the less common labels, which are the main focus of this thesis. By including both types of metrics, the evaluation makes it easier to see the balance between overall accuracy and fairness across all classes.

4.2 Baseline Experiment

Table 4.1 reports the per-label precision, recall, F1-score, and support obtained by the baseline BERT model on the test set. These results show the behaviour of the model when trained on the full (non-downsampled) dataset, before any augmentation is applied.

Overall, the model performs strongly on several high-support labels such as *love*, *amusement*, and *gratitude*, each reaching F1-scores above 0.75. These emotions

Table 4.1: Per-label performance of the baseline BERT model on the test set.

Label	**Precision**	**Recall**	**F1-score**	**Support**
love	0.831	0.726	0.775	876
amusement	0.850	0.697	0.766	1061
gratitude	0.879	0.654	0.750	1087
admiration	0.660	0.714	0.686	1981
curiosity	0.676	0.695	0.685	1272
remorse	0.828	0.511	0.603	329
joy	0.483	0.643	0.551	1156
confusion	0.475	0.655	0.551	1085
sadness	0.510	0.583	0.544	945
anger	0.471	0.630	0.539	1129
disapproval	0.460	0.629	0.531	1747
annoyance	0.409	0.742	0.527	2061
approval	0.428	0.682	0.526	2658
surprise	0.572	0.481	0.522	763
fear	0.600	0.454	0.517	449
caring	0.465	0.521	0.491	860
optimism	0.481	0.474	0.478	1231
disgust	0.386	0.531	0.447	828
desire	0.539	0.372	0.440	589
disappointment	0.343	0.596	0.436	1400
excitement	0.434	0.435	0.434	908
realization	0.317	0.452	0.373	1415
embarrassment	0.278	0.374	0.319	393
nervousness	0.287	0.290	0.289	293
relief	0.242	0.264	0.253	212
grief	0.283	0.157	0.202	108
pride	0.234	0.106	0.146	208

tend to be expressed with more explicit linguistic and semantic signals, making them easier for BERT to learn. Labels such as *admiration*, *curiosity*, and *remorse* also achieve relatively high F1-scores, indicating that the model is able to generalize well when sufficient training examples are available and the emotional expression is consistent.

However, the results also show clear weaknesses on a subset of emotion categories. In particular, *optimism* (F1 = 0.478), *nervousness* (F1 = 0.289), *relief* (F1 = 0.253), *grief* (F1 = 0.202), and *pride* (F1 = 0.146) show substantially lower F1-scores. These labels either have very few examples (for instance, *grief* appears only 108

times) or rely on subtle, ambiguous language that makes them harder for the model to predict accurately. Because they are both rare and expressed in diffuse ways, the model struggles with them even when trained on the full dataset.

This first evaluation sets the baseline performance for all later augmentation experiments. It also clearly identifies which labels need the most improvement that will be the focus of the downsampling and augmentation steps.

4.3 Downsampling Experiments

To understand how performance drops when less data is available, the training dataset was gradually downsampled in steps of 10% up to 90%. At each reduction level, a new BERT base model was trained from scratch and evaluated using the same validation and test splits. Figure 4.1 summarizes the relationship between training data size and classification performance, expressed through the Micro-F1 and Macro-F1 scores.

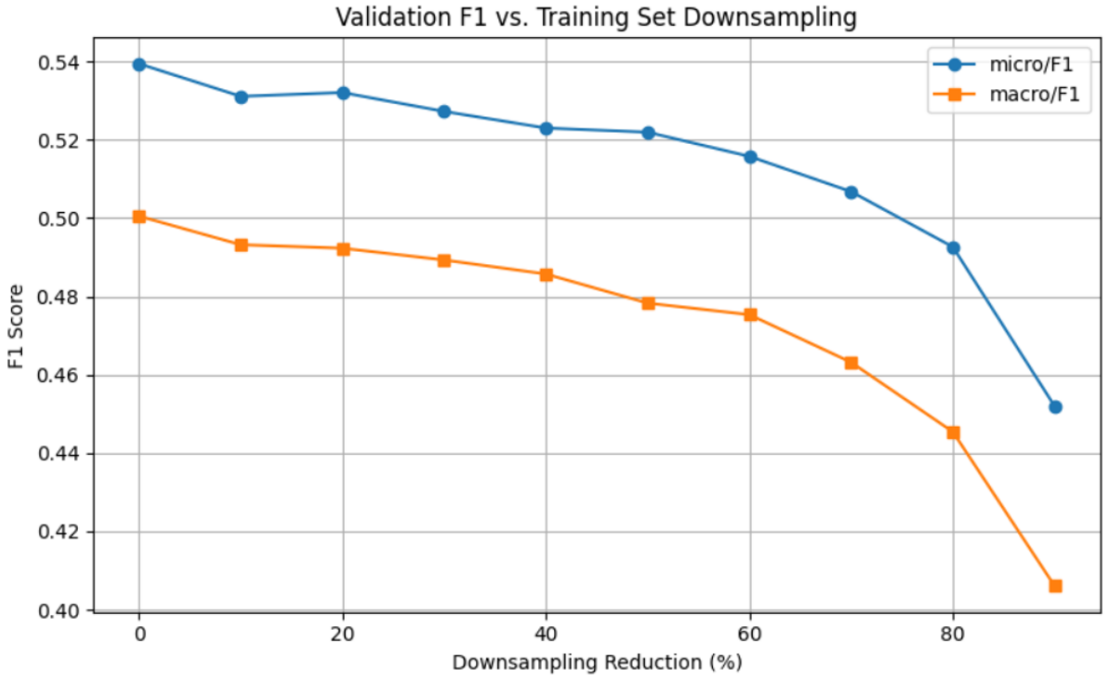

Figure 4.1: Validation F1-score versus training set downsampling. Both micro- and macro-averaged F1-scores are shown as the percentage of removed training data increases.

As expected, model performance consistently declines as the amount of available training data decreases. The **Micro-F1 score**, which aggregates contributions from

all classes proportionally to their frequency, shows a gradual decrease, indicating an overall reduction in predictive accuracy across the dataset. The **Macro-F1 score** which averages performance equally across all labels drops more sharply. This gap shows that rare emotions are more severely impacted by data scarcity than frequent ones. When the dataset is reduced to 40% of its original size (a 60% downsampling rate), the Macro-F1 score drops sharply, confirming that underrepresented classes lose performance disproportionately.

These results align with prior studies on imbalanced and low-resource NLP tasks, which shows that when models are trained on limited data, they often overfit to the majority patterns while failing to represent minority categories [15, 42]. The sharper decline of the Macro-F1 score shows that the model is becoming less able to generalize across the full range of emotions, highlighting the importance of targeted strategies such as data augmentation to better represent infrequent emotions. This observation motivated the following experiments, which apply traditional and LLM-based augmentation techniques to compensate for the effects of label scarcity.

To analyse the effect of training data scarcity, we choose the experiment where the training set for each emotion was reduced by 60%, and the baseline BERT model was trained again from scratch. The evaluation was performed on the unchanged test set to ensure comparability with the full-data baseline. Table 4.2 presents the per-class precision, recall, F1-score, and support.

In total, the results show a clear decrease in performance compared to the full-data model, confirming that BERT is strongly dependent on the availability of sufficient examples for multi-label emotion classification. High-frequency emotions such as *love*, *amusement*, and *gratitude* remain relatively stable, preserving F1-scores between 0.74 and 0.76. These labels are frequent and often expressed with distinctive lexical signals, which helps the model remain robust even when the training data is heavily reduced.

However, the reduction affects mid frequency and low frequency labels. Several medium support emotions such as *disapproval* (F1 = 0.528), *annoyance* (F1 = 0.523), *anger* (F1 = 0.521), and *approval* (F1 = 0.512) show noticeable declines. These categories are more semantically diffuse, and their expression depends on subtle context, making them particularly sensitive to data reduction.

The most noticeable declines occur in low support emotions. Labels such as *optimism* (F1 = 0.471), *caring* (F1 = 0.461), *disgust* (F1 = 0.446), *desire* (F1 = 0.449), *excitement* (F1 = 0.419), and *realization* (F1 = 0.346) now fall well below 0.50 F1, showing that the model struggles to form stable decision boundaries with limited training examples.

At the bottom of the performance distribution are the rarest emotions, which collapse almost entirely. *Relief* drops to 0.192 F1, *nervousness* to 0.234 F1, *embarrassment* to 0.254 F1, while *grief* (0.159) and *pride* (0.094) fall to near-zero

F1 despite unchanged test-time support. This pattern shows a known phenomenon in imbalanced multi-label classification which is when support becomes too low, the model defaults to majority-class predictions, severely harming the recall of rare labels.

These findings provide motivation for the augmentation strategies explored in the next sections. The big decrease of minority emotions under 60% downsampling makes them ideal targets for data augmentat, and the regression-based analysis performed later aligns with these observations.

Table 4.2: Per-label performance of the baseline model after 60% downsampling of the training data.

Emotion	**Precision**	**Recall**	**F1-score**	**Support**
love	0.758	0.771	0.764	876
amusement	0.889	0.702	0.752	1061
gratitude	0.852	0.664	0.747	1087
admiration	0.658	0.699	0.678	1981
curiosity	0.697	0.635	0.665	1272
remorse	0.773	0.486	0.597	329
confusion	0.464	0.643	0.540	1085
sadness	0.522	0.647	0.534	945
disapproval	0.464	0.611	0.528	1747
joy	0.480	0.574	0.523	1156
annoyance	0.418	0.610	0.523	2061
anger	0.454	0.610	0.521	1129
approval	0.420	0.653	0.512	2658
surprise	0.507	0.499	0.503	763
fear	0.578	0.428	0.492	449
optimism	0.448	0.496	0.471	1231
caring	0.459	0.464	0.461	860
desire	0.464	0.435	0.449	589
disgust	0.384	0.531	0.446	828
disappointment	0.338	0.595	0.431	1400
excitement	0.443	0.398	0.419	908
realization	0.317	0.452	0.346	1415
embarrassment	0.293	0.224	0.254	393
nervousness	0.294	0.195	0.234	293
relief	0.191	0.193	0.192	212
grief	0.124	0.222	0.159	108
pride	0.069	0.149	0.094	208

4.3.1 Identifying Underperforming Labels

After downsampling the training set by 60%, we performed a regression based diagnostic analysis to determine which emotion labels were affected by the reduction in data. The aim of this step is to establish a quantitative method for selecting the labels that should be targeted during data augmentation.

Figure 4.2: Identifying underperforming labels using regression based diagnostic analysis.

Figure 4.2 visualizes the relationship between each label's validation F1-score and its training support. Each point corresponds to one of the 27 emotion categories. The horizontal axis shows the number of training examples available for that label after downsampling, while the vertical axis reports its resulting F1-score. A linear regression line is fitted across all points to model the expected level of performance as a function of support size.

Labels positioned below the regression line are considered *underperforming*: their F1-scores are lower than what the model would typically achieve given the amount of training data available. Labels above the line are considered *overperforming*, indicating that the model is able to learn them effectively even with limited support.

This analysis reveals a clear pattern. Several emotions with relatively low support fall below the regression line, including *grief*, *pride*, *relief*, *nervousness*, *embarrassment*, and *desire*. These labels achieve F1-scores far below what would

be expected from their sample size, which suggests that the model struggles to generalize their emotional content. Other labels with comparable support (such as *love* or *gratitude*) are located above the regression line, indicating more better learnability.

By analyzing performance relative to expected behavior rather than raw support alone, this regression-based method provides a more reliable and data-driven approach to label selection. It identifies not only emotions that are rare in the training set, but also those whose linguistic or emotional signals may be inherently more difficult for the model to capture. The labels identified in this step which are in total 15 (*grief*, *pride*, *relief*, *nervousness*, *embarrassment*, *fear*, *desire*, *caring*, *excitement*, *optimism*, *disappointment*, *realization*, *disapproval*, *annoyance*, and *approval*) form the target set for the augmentation strategies analysed in the next sections.

4.4 Traditional Augmentation (Synonym Replacement)

The first augmentation strategy evaluated in this thesis is synonym replacement, inspired by the Easy Data Augmentation (EDA) framework. The goal was to enrich the underperforming labels identified through regression analysis by generating additional training examples that preserved the original meaning while introducing lexical variety.

The method operates by selecting one or more content words in a sentence and replacing them with WordNet-derived synonyms as mentioned in section 3.2.5. This produces new variants that remain semantically aligned with the original sentence but differ in surface form. Compared to more advanced LLM-based approaches, synonym replacement is simple, fast, and computationally inexpensive.

In this work, synonym replacement was applied *only* to the set of underperforming labels from the 60% downsampled experiment. For each target label, new examples were generated until the label's support increased toward the level of the most frequent emotion. The most frequent emotion was *approval*, with *3166* instances. Even though *approval* was identified by the regression technique as an underperforming label, it was not augmented. Each of the other 14 labels was augmented individually in separate training runs: for every run, only one label at a time was augmented up to *3166* instances, the model was retrained from scratch, and the results were recorded. This allows us to specifically examine the effect of augmenting each label, for example, evaluating performance changes when only *embarrassment* was augmented, and so on. After augmentation, standard threshold tuning was applied and evaluation was performed on the unchanged test set.

Table 4.3 reports the per-label performance after applying synonym replacement.

While some labels experienced a modest improvement in F1-score, the overall gains remained limited. In several cases, performance remained nearly unchanged or even declined slightly. This outcome is consistent with observations in prior work where synonym replacement can unintentionally introduce semantic drift when a substituted synonym does not fit the context, and it often produces unnatural phrasing. Such distortions may confuse the model or reduce the utility of the generated examples.

A closer look reveals that improvements occurred primarily in labels with slightly higher support within the underperforming group. For example, *annoyance* reaches an F1 of 0.522, close to the 60%-downsampled baseline. Similarly, *disapproval* achieves 0.516 and *fear* reaches 0.476. Only pride reached an F1 higher that the downsampled baseline. These gains suggest that synonym replacement can help when the semantic space of a label is relatively broad and tolerant to lexical variation.

However, for the rarest labels, such as *grief*, and *relief* the improvements are minimal. These emotions are typically expressed with specific, nuanced language, and synonym substitution often fails to preserve the intended emotional meaning, leading to limited benefit during model training.

Although synonym replacement provides a good baseline and can modestly provide an improvement, its impact is inconsistent and weak. These leads us to explore LLM-based techniques of generative methods in the following section.

Table 4.3: Per-label performance after traditional synonym-replacement augmentation.

Emotion	**Precision**	**Recall**	**F1-score**	**Support**
nervousness	0.234	0.222	0.228	293
optimism	0.428	0.461	0.444	1231
annoyance	0.434	0.656	0.522	2061
disapproval	0.432	0.642	0.516	1747
excitement	0.381	0.424	0.401	908
grief	0.150	0.157	0.154	108
disappointment	0.340	0.538	0.416	1400
desire	0.337	0.458	0.388	589
realization	0.246	0.559	0.342	1415
fear	0.513	0.443	0.476	449
embarrassment	0.168	0.407	0.238	393
relief	0.163	0.226	0.189	212
pride	0.115	0.216	**0.150**	208
caring	0.383	0.505	0.436	860

4.5 LLM-based Data Generation

After establishing the baseline and evaluating the traditional synonym replacement approach, this section presents the results of the second augmentation strategy which is large language model based data generation. While the previous chapter introduced the methodology and implementation details of this approach, here we focus on the results and their interpretation.

The goal of this strategy was to improve performance for underperforming emotion labels by generating high-quality synthetic examples that preserved both meaning and emotional tone. Instead of replacing individual words with synonyms, this approach used a generative model to rewrite entire sentences, thereby achieving richer lexical and syntactic diversity as mentioned in section 3.2.5. Same as in the previous section, for each target label, new examples were generated until the label's support increased toward the level of the most frequent emotion. For every run, only one label at a time was augmented up to *3166* instances, the model was retrained from scratch, and the results were recorded.

For this purpose, the `Mistral-7B-Instruct` model [37] was used. This instruction tuned large language model, containing approximately 7 billion parameters, is trained to follow human-like instructions and produce contextually appropriate outputs. Its generative ability allows it to produce paraphrases that maintain emotional fidelity while introducing natural linguistic variety.

Two prompting strategies were tested and compared: (1) a *simple prompt*, which focused on minimal rewriting using synonyms, and (2) an *advanced prompt*, which provided richer guidance, stylistic constraints, and in-context examples. The comparison between these two prompts shows the importance of prompt engineering in LLM behavior and achieving consistent label augmentations.

4.5.1 Simple Prompt

The initial experiments employed a straightforward prompting strategy, which instructed the model to perform synonym-based rewriting for a given emotion label. The prompt template was:

> *"Rewrite the following sentence using synonyms. Keep the same meaning and emotion '{label}': '{sentence}'. Respond with one sentence only."*

This configuration led to modest gains, with improvements observed for only 6 out of 14 targeted labels, specifically *relief, embarrassment, fear, desire, grief,* and *nervousness*. The limited improvement can be attributed to two main factors. First, the simplicity of the prompt provided little contextual guidance, leading the model to produce literal synonym substitutions rather than meaningful paraphrases. Second, the lack of stylistic and semantic constraints occasionally introduced subtle

shifts in tone or emotion, a phenomenon known as semantic drift. As a result, some generated sentences were linguistically valid but failed to express the intended emotional meaning, and so reducing their effectiveness as augmentation samples.

These findings are consistent with earlier work on data augmentation, which has shown that simple word-level transformations often provide limited benefits for tasks that rely on subtle and consistent semantic meaning [33].

Table 4.4: Per-label performance after LLM-based augmentation with the simple prompt.

Emotion	Precision	Recall	F1-score	Support
nervousness	0.152	0.244	**0.187**	293
optimism	0.384	0.481	0.427	1231
annoyance	0.412	0.599	0.489	2061
disapproval	0.412	0.553	0.472	1747
excitement	0.286	0.414	0.338	908
grief	0.123	0.147	**0.134**	108
disappointment	0.325	0.510	0.400	1400
desire	0.397	0.367	**0.382**	589
realization	0.199	0.539	0.291	1415
fear	0.413	0.398	**0.405**	449
embarrassment	0.133	0.381	**0.198**	393
relief	0.180	0.202	**0.191**	212
pride	0.074	0.143	0.097	208
caring	0.337	0.379	0.357	860

4.5.2 Advanced Prompt

To overcome these limitations, a more context-rich and instructive prompt was developed. This new version explicitly described the rewriting task, constrained style, and provided multiple examples of desired behavior. The final prompt used for training the LLM was:

> *You are a helpful assistant that rewrites sentences for social media.*
> *You receive a sentence and an emotion. You have to rewrite the sentence preserving the original meaning and emotion.*
> *Do not add hashtags, emojis, or any new content unless already present.*
> *Keep the same text style.*
> *Return only one sentence.*
>
> *Examples:*

Original: I just finished my first half marathon.
Rewritten: Just completed my first half marathon!

Original: So grateful to all my colleagues for pulling this off.
Rewritten: Huge thanks to my colleagues for making this happen.

Original: Wrapping up a great quarter with an amazing team.
Rewritten: Finishing a fantastic quarter alongside a great team.

This advanced prompt produced consistently better results, leading to improvements in 8 out of 14 targeted labels. Providing richer context and adding examples helped the model better understand how each sentence should be rewritten and where the emotional boundaries of each label lie. The generated sentences ended up sounding more fluent and natural, and they also stayed closer to the intended emotion, with fewer instances of labels drifting away from their original meaning.

By supplying structured examples, the prompt guided the model toward producing emotionally consistent variations, which led to clear improvements in macro-level F1 compared to both the simpler prompt and the traditional synonym-replacement approach.

Overall, the results show that although LLM-based augmentation requires more computational resources than rule-based methods, it generates synthetic data that more accurately reflects the emotional patterns in the dataset. This makes it a good option for addressing label imbalance in multi-label emotion classification tasks.

An important outcome of the methodological exploration was the realization that the quality of the prompt critically influences the quality of the generated data. The simpler prompt that was introduced in the previous section produced only modest improvements. In contrast, the advanced prompt offered richer context and clearer constraints, including stylistic guidance, reminders not to introduce new content, and examples showing how to produce faithful paraphrases. This more explicit setup resulted in synthetic sentences that were not only more coherent but also more accurately reflected the intended emotion.

The augmented dataset was then used to re-train the BERT model, followed by threshold tuning and evaluation on the original test set. The resulting per-label metrics are reported in Table 4.5.

The performance gains are noticeably larger and more consistent than those obtained with synonym replacement. Out of the 14 targeted labels, 8 showed an improvement in F1-score relative to the downsampled baseline, confirming the effectiveness of the LLM-based method. Notable improvements include:

- **grief**: F1 increases from 0.159 to 0.192,
- **relief**: from 0.192 to 0.240,

- **embarrassment**: from 0.234 to 0.302,
- **realization**: from 0.346 to 0.366,
- **caring**: from 0.461 to 0.455 (comparable but stable),
- **fear**: from 0.492 to 0.493,
- **optimism**: from 0.471 to 0.451 (minor decline),
- **annoyance**, **disapproval**, and **excitement** remain strong performers with stable F1.

While not all labels improved, the overall trend is clear. LLM-generated examples exhibit higher semantic fidelity and create richer training signals than synonym substitution. Particularly for subtle emotions with low support (*grief*, *pride*, and *relief*) LLM rewriting offers a significant advantage by producing naturalistic paraphrases that better capture the underlying affective meaning. So, we can see that augmentation quality, not just quantity, plays an important role in improving model performance for minority labels.

Table 4.5: Per-label performance after LLM-based augmentation with the advanced prompt.

Emotion	Precision	Recall	F1-score	Support
nervousness	0.258	0.259	**0.259**	293
optimism	0.450	0.453	0.451	1231
annoyance	0.429	0.649	0.517	2061
disapproval	0.433	0.669	0.526	1747
excitement	0.380	0.479	**0.424**	908
grief	0.174	0.213	**0.192**	108
disappointment	0.351	0.538	0.425	1400
desire	0.407	0.433	0.420	589
realization	0.275	0.548	**0.366**	1415
fear	0.563	0.439	**0.493**	449
embarrassment	0.316	0.290	**0.302**	393
relief	0.208	0.283	**0.240**	212
pride	0.188	0.130	**0.153**	208
caring	0.393	0.540	0.455	860

4.6 Comparison

Table 4.6: Performance comparison for all 14 emotion labels across techniques.

Label	Technique	F1	Precision	Recall
nervousness	Original	0.303	0.356	0.263
	Downsampled	0.234	0.294	0.195
	Traditional Augmentation	0.228	0.234	0.222
	LLM-based	**0.259**	0.258	0.259
excitement	Original	0.432	0.444	0.421
	Downsampled	0.419	0.443	0.398
	Traditional Augmentation	0.401	0.381	0.424
	LLM-based	**0.424**	0.380	0.479
grief	Original	0.130	0.115	0.148
	Downsampled	0.159	0.124	0.222
	Traditional Augmentation	0.154	0.150	0.157
	LLM-based	**0.192**	0.174	0.213
realization	Original	0.387	0.293	0.568
	Downsampled	0.346	0.246	0.584
	Traditional Augmentation	0.342	0.246	0.559
	LLM-based	**0.366**	0.275	0.548
fear	Original	0.532	0.613	0.470
	Downsampled	0.492	0.578	0.428
	Traditional Augmentation	0.476	0.513	0.443
	LLM-based	**0.493**	0.563	0.439
embarrassment	Original	0.304	0.265	0.356
	Downsampled	0.254	0.293	0.224
	Traditional Augmentation	0.238	0.168	0.407
	LLM-based	**0.302**	0.316	0.290
relief	Original	0.235	0.218	0.255
	Downsampled	0.192	0.191	0.193
	Traditional Augmentation	0.189	0.163	0.226
	LLM-based	0.240	0.208	0.283
pride	Original	0.124	0.098	0.168
	Downsampled	0.094	0.069	0.149
	Traditional Augmentation	0.150	0.115	0.216
	LLM-based	0.153	0.188	0.130
optimism	Original	0.488	0.499	0.477
	Downsampled	0.471	0.448	0.496
	Traditional Augmentation	0.444	0.428	0.461
	LLM-based	0.451	0.450	0.453

Continued on next page

Label	Technique	F1	Precision	Recall
annoyance	Original	0.529	0.433	0.681
	Downsampled	0.522	0.418	0.695
	Traditional Augmentation	0.522	0.434	0.656
	LLM-based	0.517	0.429	0.649
disapproval	Original	0.545	0.496	0.604
	Downsampled	0.528	0.464	0.611
	Traditional Augmentation	0.516	0.432	0.642
	LLM-based	0.526	0.433	0.669
disappointment	Original	0.454	0.371	0.586
	Downsampled	0.431	0.338	0.595
	Traditional Augmentation	0.416	0.340	0.538
	LLM-based	0.425	0.351	0.538
desire	Original	0.456	0.514	0.409
	Downsampled	0.449	0.464	0.435
	Traditional Augmentation	0.388	0.337	0.458
	LLM-based	0.420	0.407	0.433
caring	Original	0.500	0.443	0.574
	Downsampled	0.461	0.459	0.464
	Traditional Augmentation	0.436	0.383	0.505
	LLM-based	0.455	0.393	0.540

In this section we will compare all the techniques that were used. As already mentioned in the previous sections these techniques are: the baseline model trained on the full dataset, a version trained on a dataset downsampled by 60% per label, a model trained with traditional synonym-based augmentation, and a model trained with LLM-based augmentation using Mistral-7B-Instruct (the advanced prompt). These configurations can show a systematic assessment of how different augmentation strategies influence both overall performance and label specific recovery under class imbalance.

Baseline vs. Downsampled (60% Reduction). As expected, reducing the support for each emotion label by 60% resulted in a noticeable decrease of performance across most classes. High-frequency emotions such as *love*, *gratitude*, and *amusement* remained relatively stable, whereas mid-frequency labels (e.g., *disapproval*, *annoyance*, *optimism*) experienced moderate declines. The most severe decline occurred among low-support labels such as *grief*, *pride*, *nervousness*, and *relief*, which in some cases lost more than half of their original F1 score. These results can show the well-known sensitivity of minority classes to data reduction and highlight the need for augmentation approaches for the low-resource labels.

Traditional Synonym Replacement. Applying synonym-replacement augmentation produced mixed results. Only 1 out of the 14 targeted labels showed an improvement. This label is pride which the F1-score for the downsampled technique is *0.094* and for the traditional synonym replacement is *0.150*. For several labels, performance decreased relative to the downsampled model, largely due to lexical substitutions that disrupted the contextual or emotional meaning of the original text. Simple word-level transformations often introduce semantic drift and yield unnatural or stylistically inconsistent sentences. This method provides a useful baseline but lacks the expressive power needed to reliably improve minority emotion categories.

LLM-Based Augmentation. In contrast, augmenting the data with the model `Mistral-7B-Instruct` yielded the most improvements. Out of the 14 targeted labels, 8 exhibited higher F1 scores after augmentation. Gains were especially pronounced for low-frequency emotions such as *embarrassment*, *relief*, *pride*, and *grief*, where synthetic examples helped compensate for data scarcity. For *nervousness*, the F1-score increases slightly from 0.234 to 0.259 because recall improves significantly (0.195 to 0.259), even though precision drops a little (0.294 to 0.258). In *excitement*, the improvement in F1-score from 0.419 to 0.424 is driven by a strong gain in recall (0.398 to 0.479), despite a noticeable decrease in precision (0.443 to 0.380). For *grief*, all three metrics improve: F1-score rises from 0.159 to 0.192, precision from 0.124 to 0.174, and recall from 0.222 to 0.213, although recall only changes slightly. In the case of *realization*, the F1-score goes up from 0.346 to 0.366 mainly due to better precision (0.246 to 0.275), while recall declines slightly (0.584 to 0.548), showing a trade-off. For *fear*, the F1-score barely changes (0.492 to 0.493) because precision decreases slightly (0.578 to 0.563) while recall improves modestly (0.428 to 0.439), keeping the overall balance similar. Both *embarrassment* and *relief* show clear improvements across all metrics: for embarrassment, F1-score rises from 0.254 to 0.302, precision from 0.293 to 0.316, and recall from 0.224 to 0.290; for relief, F1-score improves from 0.192 to 0.240, precision from 0.191 to 0.208, and recall from 0.193 to 0.283. Finally, *pride* stands out for its big jump in precision (0.069 to 0.188), which drives the F1-score up from 0.094 to 0.153, even though recall falls slightly (0.149 to 0.130). This means the model is more selective but still performs better overall.

The quality of LLM-generated paraphrases played the most important role where the advanced prompt ensured that rewrites remained faithful to the original meaning and emotional tone while introducing natural linguistic variability. This led to semantically coherent and stylistically correct examples, improving the model's ability to generalize to rare emotional expressions.

Overall Comparison. A summary of the outcomes across conditions is shown below:

- **Downsampled (60%)**: Performance declines across most labels where minority classes heavily affected.
- **Traditional augmentation**: Improvements in 1/14 labels where small gains and occasional degradation.
- **LLM-based augmentation**: Improvements in 8/14 labels where stronger gains, especially for low-support emotions.

The results show that LLM-based augmentation is the most effective approach for restoring performance when dealing with severe class imbalance. It not only outperforms traditional synonym replacement but also delivers larger and more consistent gains. Its ability to generate text that feels emotionally accurate and stylistically natural makes it especially valuable for enriching minority emotion classes in multi-label classification tasks.

Overall, the comparison supports the main claim of this thesis: high-quality synthetic data generated by large language models can significantly improve classification performance when data is limited. LLM-based augmentation stands out as clearly superior to traditional methods, providing both stronger quantitative results and more meaningful training examples.

Chapter 5

Discussion

This chapter reflects on the main findings of the experimental study and connects them back to the research questions posed in the introduction. The discussion is organised around four themes: how the baseline model behaves under different levels of data scarcity, the role of threshold tuning and regression-based analysis in understanding label performance, the comparative impact of traditional and LLM-based augmentation, and the quality, limitations, and broader implications of using synthetic data for multi-label emotion classification.

The downsampling experiments clearly show how strongly the model depends on the availability of training data. As the proportion of available data decreases, both micro- and macro-averaged F1-scores decline in a smooth and predictable manner. The drop is steeper for macro-F1 than for micro-F1, which confirms that minority labels are disproportionately affected by scarcity. This behaviour is consistent with how the metrics are defined: micro-F1 is dominated by frequent labels, whereas macro-F1 gives each label equal weight and therefore exposes failures on rare classes more clearly. At higher reduction levels, such as the 60 percent setting used throughout this work, several minority emotions fall to very low F1 values even though their test-time support remains unchanged.

The downsampling study also shows how precision and recall shift when data become limited. For low-support labels, the model tends to behave more conservatively where precision often remains relatively high, but recall drops sharply. In other words, when the model predicts a rare emotion it is likely to be correct, but it tends to make such predictions too rarely. For emotion recognition tasks, this conservative behaviour may be undesirable because it prioritises avoiding false positives over detecting subtle emotional cues. Depending on the application, it might be necessary to bias the system toward higher recall for particular emotions, especially those with high practical relevance.

Threshold tuning can be used as an effective method for improving performance under these circumstances. Tuning one threshold per emotion on the validation set

improves both micro- and macro-F1 scores, with clear benefits for underrepresented labels. By adapting each decision boundary to the empirical score distribution of the corresponding label, threshold tuning compensates for skewed priors and differences in difficulty across emotions. However, it is important to recognise that this technique operates only at the decoding stage and does not influence what the model has learned internally. When the training signal for an emotion is extremely limited, even the best choice of threshold cannot recover information that the model never acquired. As a result, the benefit of threshold tuning gradually diminishes as downsampling becomes more aggressive.

To better understand where the model struggles, a regression-based analysis was performed. By regressing per-label F1-scores on label support and studying the residuals, it becomes possible to distinguish labels that perform poorly simply because they are rare from those that perform worse than expected given their support. Emotions that fall clearly below the regression line, such as grief, pride, relief, and nervousness, appear to be intrinsically difficult. Their difficulty may arise from linguistic subtlety, context-dependent meaning, or diffuse emotional expression. This diagnostic perspective proved important for guiding the augmentation stage, ensuring that new examples were added where they could make the most meaningful difference rather than being allocated purely by counting frequencies.

The comparison of augmentation strategies shows that not all strategies provide the same benefit. Traditional synonym replacement, inspired by Easy Data Augmentation, provides a useful baseline. When applied only to the labels identified as underperforming, it increases lexical diversity at very low computational cost and sometimes recovers a small amount of performance, mainly by increasing recall. However, the overall improvements are modest and often inconsistent. Only one of the fourteen targeted labels, *pride*, achieves a clear improvement over the downsampled baseline. This limited success aligns with what can be observed qualitatively. Even when synonyms are technically correct, they often feel slightly unnatural or shift the emotional tone in subtle ways. For tasks that depend on fine-grained emotional signals, even small mismatches can reduce the usefulness of the new examples.

The LLM-based augmentation using the `Mistral-7B-Instruct` model presents a more promising alternative. When paired with the advanced prompt that includes clear instructions and in-context examples, the model generates paraphrases that are natural, stylistically consistent with Reddit data, and faithful to the original emotional label. These synthetic sentences are not simple word substitutions but full rewrites that preserve meaning and emotional tone. Quantitatively, this results in improvements for eight of the fourteen targeted labels, with particularly strong gains among the most data-scarce emotions such as embarrassment, relief, grief, and pride. Although the absolute F1 values for these labels remain low, as expected for extremely rare emotions, the relative improvements over both the downsampled

and synonym-augmented models are significant.

One of the clearest findings from the augmentation experiments is the importance of prompt design. The initial, simpler prompt produced only modest gains because it offered little guidance beyond requesting synonym-based rewriting. The advanced prompt, by contrast, provided explicit constraints, clear instructions, and examples that demonstrated the desired behaviour. This richer formulation helped the behaviour of the LLM and reduce semantic drift, ensuring that the model generated paraphrases that remained faithful to the intended emotion.

The quality of augmented data therefore becomes an important factor in determining the effectiveness of any augmentation strategy. For synonym replacement, quality was controlled by limiting replacement rates and filtering unnatural outputs. For the LLM-based generation, quality control involved careful prompt crafting, appropriate generation parameters such as temperature and top-p sampling, and post-processing to remove duplicates or malformed sentences. Augmentation should prioritise semantic fidelity and domain alignment, particularly for subtle tasks such as emotion classification.

Despite the improvements achieved, several types of errors persist. The model frequently confuses emotions with similar valence, such as sadness and disappointment or anger and disgust, and these confusions become more pronounced as data decrease. The model also struggles with co-occurring emotions in multi-label settings, often predicting only one emotion when human annotators indicated several. This can show that independent label prediction may be insufficient for capturing complex emotional dependencies. Finally, the model continues to struggle with sarcastic or ironic comments, which require contextual meaning that are not present in isolated sentences. While augmentation improves coverage of surface-level patterns, it does not address this deeper contextual challenge.

This study also comes with limitations. The unanimity-based preprocessing yields high-precision labels but excludes cases where annotators disagreed, potentially removing subtle or ambiguous emotional expressions. The downsampling procedure treats all labels equally and does not account for differences in inherent difficulty or stylistic variation across subreddits. The regression assumes a linear relationship between support and performance, which may be an oversimplification in extreme cases. Moreover, all experiments were conducted in English and on Reddit data, making it unclear how well the findings transfer to other languages, platforms, or model architectures.

From a practical perspective, the results point to several recommendations. Threshold tuning should be included by default in multi-label classification pipelines. Augmentation should boost labels that underperform, not just those that appear infrequently. Traditional augmentation can serve as a quick baseline, but LLM-based augmentation is more effective when computational resources allow, especially when prompts are carefully designed and high-quality constraints are applied.

Throughout the experiments, reproducibility was prioritised. The workflow uses fixed data splits, deterministic preprocessing, seeded randomness, and full model reinitialisation for each setting. The findings were most sensitive to the level of downsampling, the augmentation target counts, and the constraints used in the LLM prompt.

Finally, synthetic data generation raises ethical considerations. Even when the task is restricted to rewriting existing sentences without adding new content, there is a potential risk of amplifying biases or introducing stylistic patterns that distort the data distribution. In this thesis, these risks were mitigated by constraining the generation process, avoiding content creation, and evaluating exclusively on human-written test data.

Chapter 6

Future Work

The findings of this thesis open several promising directions for future research, both in terms of methodological refinement and broader applications. While the experiments show the value of targeted augmentation, particularly when using large language models, there are still open questions about how to further improve data quality, model robustness, and real-world applicability in imbalanced multi-label emotion classification.

A first area for future exploration concerns the selection of augmentation targets. In this thesis, labels were chosen based on a regression-based underperformance analysis, which proved effective in identifying emotions that are difficult to learn. However, the method could be enriched by incorporating additional criteria such as semantic diversity, lexical variability, or context-specific ambiguity. Combining these measures with performance-based diagnostics could lead to a more complete understanding of label difficulty and support more precise augmentation strategies. An adaptive selection mechanism, where labels are re-evaluated throughout training, may also help refine augmentation by responding to the model's evolving performance.

Another direction is the selection of the most representative sentences for each label and using these sentences as templates for generating new synthetic examples. By identifying prototypical instances that capture the core semantics of each emotion, augmentation could produce more coherent and contextually relevant synthetic data. This approach would not only improve the quality of generated samples but also help maintain consistency across augmented datasets, ultimately enhancing model generalization.

A second direction involves modelling label dependencies more explicitly. The current architecture treats emotion labels as independent outputs, despite the fact that many emotions tend to co-occur or rely on similar linguistic meanings. Future work could explore hierarchical, graph-based, or correlation-aware models that make use of observed co-occurrence patterns. Approaches such as classifier

chains, conditional dependency networks, or transformers with structured output spaces may help the model capture these relationships more effectively, reducing common errors such as missing secondary emotions or confusing emotions with similar meaning.

There is also space for improvement in synthetic data generation. One promising direction is the integration of human-in-the-loop validation or reinforcement learning from human feedback, where generated paraphrases are iteratively refined against explicit quality criteria. This could help issues such as subtle semantic drift or stylistic inconsistencies. Another valuable extension is multilingual augmentation. As instruction-tuned models are increasingly available in multiple languages, applying the methods developed here to multilingual or cross-lingual emotion datasets could significantly broaden the reach of the approach. Additionally, automatic prompt optimisation using techniques such as genetic search, gradient-free optimisation, or prompt tuning could reduce dependence on manual prompt crafting and help produce more consistent synthetic examples.

Hybrid augmentation strategies also represent an interesting direction to explore. Rather than relying only on synonym replacement or model-based rewriting, future research could examine how different augmentation methods interact. For instance, lexical transformations can increase surface-level variability while large language model paraphrasing helps preserve meaning. Combining these approaches may produce richer and more balanced training data. Likewise, context-aware generation, where the model has access to extended conversation history or subreddit metadata, could be useful when emotion depend on broader discussion or social context.

Evaluation frameworks can be expanded as well. Beyond micro and macro F1 scores, future studies could consider calibration metrics, or label correlation aware scores that better capture the structure of multi-label predictions. These metrics may offer insights into how augmentation affects prediction confidence and reliability, particularly for less frequent emotions.

To sum up, future work can build on the contributions of this thesis by improving augmentation quality, advancing model architectures, broadening cross-lingual applications, and addressing the challenges of real-world deployment. These directions point toward a more flexible, and contextually aware framework for emotion classification in imbalanced multi-label settings.

Chapter 7

Conclusion

This thesis set out to explore how targeted data augmentation can help address the challenges created by class imbalance in multi-label emotion classification. By combining controlled data reduction, systematic diagnostics, and synthetic data generation, the work shows that carefully designed augmentation strategies, especially those that rely on large language models, offer an effective way to strengthen performance for rare and difficult emotion categories.

The study began by establishing a strong baseline using a BERT-based classifier trained on a cleaned and consolidated version of the GoEmotions dataset. Controlled downsampling experiments revealed how performance declines when label support decreases, particularly for subtle and infrequent emotions. These findings showed the structural sensitivity of multi-label models to imbalance and underscored the need for augmentation methods that do more than simply increase sample size.

An important contribution of this work is the regression-based diagnostic analysis used to identify labels that underperform. Instead of selecting augmentation targets only by their frequency, this method identifies labels that perform worse than expected given their level of support. This approach therefore offers a principled and generalizable way to prioritize which labels should receive augmentation.

The experimental comparison between augmentation strategies showed clear differences in their impact. Traditional synonym replacement served as a useful baseline but produced limited and sometimes inconsistent gains, often due to contextual drift caused by isolated word substitutions. In contrast, augmentation based on large language models, using the *Mistral 7B model*, produced stronger and more reliable improvements, particularly when paired with a prompt designed to preserve emotional accuracy and stylistic consistency. Augmentation quality, especially in terms of semantic precision and natural expression, plays a central role in improving classifier performance for rare emotional categories.

More broadly, the thesis show how the interaction between diagnostic analysis, model calibration, and high-quality augmentation creates a flexible pipeline for

addressing imbalance. The modular structure of the implementation allows the approach to be extended with new generative models, prompt designs, or selection criteria, making it suitable both for future research and for practical applications.

In summary, this thesis shows how modern generative tools can help with multi-label emotion classification, especially when some emotion classes are underrepresented. By combining statistical analysis with high-quality synthetic text generation, it provides a clear and flexible approach for improving the detection of minority emotions.

Bibliography

[1] Bohan Li, Yutai Hou, and Wanxiang Che. «Data augmentation approaches in natural language processing: A survey». In: *AI Open* 3 (2022), pp. 71–90. ISSN: 2666-6510. DOI: 10.1016/j.aiopen.2022.03.001. URL: http://dx.doi.org/10.1016/j.aiopen.2022.03.001 (cit. on pp. 1, 11, 15).

[2] Dorottya Demszky et al. «GoEmotions: A Dataset of Fine-Grained Emotions». In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020). URL: https://arxiv.org/abs/2005.00547 (cit. on pp. 3, 13–15).

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423 (cit. on pp. 3, 7, 15, 20, 23, 26, 27).

[4] Grigorios Tsoumakas and Ioannis Katakis. «Multi-label classification: An overview». In: *International Journal of Data Warehousing and Mining (IJDWM)* 3.3 (2007), pp. 1–13 (cit. on p. 5).

[5] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. «Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification». In: *Advances in neural information processing systems* 32 (2019) (cit. on p. 5).

[6] Shaolin Zhu, Shaoyang Xu, Haoran Sun, Leiyu Pan, Menglong Cui, Jiangcun Du, Renren Jin, António Branco, Deyi Xiong, et al. «Multilingual Large Language Models: A Systematic Survey». In: *arXiv preprint arXiv:2411.11072* (2024) (cit. on p. 5).

[7] Ling Yuan, Xinyi Xu, Ping Sun, Hai ping Yu, Yin Zhen Wei, and Jun jie Zhou. «Research of multi-label text classification based on label attention and correlation networks». In: *PloS one* 19.9 (2024), e0311305 (cit. on pp. 6, 7).

[8] Anqi Li and Lin Zhang. «Multi-Label Text Classification Based on Label-Sentence Bi-Attention Fusion Network with Multi-Level Feature Extraction». In: *Electronics* 14.1 (2025), p. 185 (cit. on pp. 6, 7).

[9] Entropy2333. *A Curated List of Papers for Multi-Label Classification*. 2023. URL: `https://github.com/entropy2333/awesome-multi-label-paper-list` (cit. on p. 7).

[10] Qunbo Wang, Hangu Zhang, Wentao Zhang, Lin Dai, Yu Liang, and Haobin Shi. «Deep active learning for multi label text classification». In: *Scientific Reports* 14.1 (2024), p. 28246 (cit. on p. 7).

[11] Wenlong Hu, Qiang Fan, Hao Yan, Xinyao Xu, Shan Huang, and Ke Zhang. «A Survey of Multi-Label Text Classification Under Few-Shot Scenarios». In: *Applied Sciences* 15.16 (2025), p. 8872 (cit. on p. 7).

[12] Haoran Luo, Tengfei Shao, Shenglei Li, and Tomoji Kishi. «An innovative 3D attention mechanism for multi-label emotion classification: H. Luo et al.» In: *Scientific Reports* 15.1 (2025), p. 35951 (cit. on pp. 7, 14, 15).

[13] Steven Y Feng, Jordan Y Park, Yu Zhai, Stephen Felcone, Aman Madaan, and Christopher Re. «A Survey of Data Augmentation Approaches for NLP». In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (2021), pp. 968–988. URL: `https://arxiv.org/abs/2105.03075` (cit. on pp. 7, 10, 12, 15, 30).

[14] Fengjun Wang, Moran Beladev, Ofri Kleinfeld, Elina Frayerman, Tal Shachar, Eran Fainman, Karen Lastmann Assaraf, Sarai Mizrachi, and Benjamin Wang. «Text2Topic: Multi-label text classification system for efficient topic detection in user generated content with zero-shot capabilities». In: *arXiv preprint arXiv:2310.14817* (2023) (cit. on pp. 7, 8, 13).

[15] Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. «A survey of methods for addressing class imbalance in deep-learning based natural language processing». In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. 2023, pp. 523–540 (cit. on pp. 8, 15, 40).

[16] Vitor Werner de Vargas, Jorge Arthur Schneider Aranda, Ricardo dos Santos Costa, Paulo Ricardo da Silva Pereira, and Jorge Luis Victória Barbosa. «Imbalanced data preprocessing techniques for machine learning: a systematic mapping study». In: *Knowledge and Information Systems* 65.1 (2023), pp. 31–57 (cit. on pp. 8, 9).

[17] 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 (cit. on p. 8).

[18] Cristian Padurariu and Mihaela Elena Breaban. «Dealing with data imbalance in text classification». In: *Procedia Computer Science* 159 (2019), pp. 736–745 (cit. on p. 8).

[19] Behnam Neyshabur Idrissi et al. «Class Imbalance in Out-of-Distribution Datasets: Improving the Robustness of the Vanilla ERM Policy». In: *arXiv preprint arXiv:2206.01601* (2022). URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9274264/ (cit. on p. 9).

[20] Zhiyuan Wang et al. «Solving Data Imbalance in Text Classification With Constructing Contrastive Samples». In: *IEEE Access* 11 (2023), pp. 84310–84321. URL: https://ieeexplore.ieee.org/document/10225302/ (cit. on p. 9).

[21] Sota Nemoto, Shunsuke Kitada, and Hitoshi Iyatomi. «Majority or minority: Data imbalance learning method for named entity recognition». In: *IEEE Access* (2024) (cit. on p. 9).

[22] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. «ADASYN: Adaptive synthetic sampling approach for imbalanced learning». In: *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. Ieee. 2008, pp. 1322–1328 (cit. on p. 9).

[23] Muhammad Mujahid, EROL Kına, Furqan Rustam, Monica Gracia Villar, Eduardo Silva Alvarado, Isabel De La Torre Diez, and Imran Ashraf. «Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering». In: *Journal of Big Data* 11.1 (2024), p. 87 (cit. on pp. 9, 10).

[24] Lwin Moe, Uyen Trang Nguyen, and Boi Trinh Luu. «Mitigating Class Imbalance in Fact-Checking Datasets Through LLM-Based Synthetic Data Generation». In: *Proceedings of the 4th ACM International Workshop on Multimedia AI against Disinformation*. 2025, pp. 73–80 (cit. on pp. 9, 30).

[25] Zihan Ke, Bing Wang, et al. «Data Augmentation for Neural NLP». In: *arXiv preprint arXiv:2302.11412* (2023). URL: https://arxiv.org/abs/2302.11412 (cit. on p. 12).

[26] Jason Wei and Kai Zou. «EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks». In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 6382–6388. URL: https://aclanthology.org/D19-1670 (cit. on pp. 12, 15, 30).

[27] Arthur Elwing Torres, Edleno Silva de Moura, Altigran Soares da Silva, Mario A Nascimento, and Filipe Mesquita. «An experimental study on data augmentation techniques for named entity recognition on low-resource domains». In: *arXiv preprint arXiv:2411.14551* (2024) (cit. on p. 12).

[28] Md Saroar Jahan, Mourad Oussalah, Djamila Romaissa Beddia, Nabil Arhab, et al. «A comprehensive study on nlp data augmentation for hate speech detection: Legacy methods, bert, and llms». In: *arXiv preprint arXiv:2404.00303* (2024) (cit. on p. 12).

[29] Yaping Chai, Haoran Xie, and Joe S Qin. «Text data augmentation for large language models: A comprehensive survey of methods, challenges, and opportunities». In: *arXiv preprint arXiv:2501.18845* (2025) (cit. on pp. 13, 14).

[30] Connor Shorten and Taghi M Khoshgoftaar. «A Comprehensive Survey on Data Augmentation». In: *arXiv preprint arXiv:2405.09591* (2025). URL: https://arxiv.org/abs/2405.09591 (cit. on p. 13).

[31] Iryna Bashynska, Mykhailo Sarafanov, and Olga Manikaeva. «Research and development of a modern deep learning model for emotional analysis management of text data». In: *Applied Sciences* 14.5 (2024), p. 1952 (cit. on p. 14).

[32] Sandhya Ramakrishnan and LD Dhinesh Babu. «Improving Multi-Label Emotion Classification on Imbalanced Social Media Data with BERT and Clipped Asymmetric Loss». In: *IEEE Access* (2025) (cit. on pp. 14, 15).

[33] Yi Wang et al. *Large Language Models on Fine-Grained Emotion Detection Dataset with Data Augmentation and Transfer Learning*. 2024. URL: https://arxiv.org/abs/2403.06108 (cit. on pp. 15, 30, 46).

[34] Zahra Ahanin, Maizatul Akmar Ismail, and Tutut Herawan. «PERFORMANCE EVALUATION OF MULTILABEL EMOTION CLASSIFICATION USING DATA AUGMENTATION TECHNIQUES». In: *Malaysian Journal of Computer Science* 37.2 (2024), pp. 154–168 (cit. on p. 15).

[35] Raj Kumar et al. «Comparative Analysis of Text-Based Emotion Detection on GoEmotions Dataset». In: *2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. 2024, pp. 1–6. URL: https://ieeexplore.ieee.org/document/10541692 (cit. on p. 15).

[36] Thomas Wolf et al. «Transformers: State-of-the-Art Natural Language Processing». In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2020, pp. 38–45 (cit. on pp. 15, 25, 26).

[37] Albert Q. Jiang et al. «Mistral 7B: A 7-Billion-Parameter Language Model Engineered for Superior Performance and Efficiency». In: *arXiv preprint arXiv:2310.06825* (2023). Version v0.1 (released Oct. 10 2023). URL: https://arxiv.org/abs/2310.06825 (cit. on pp. 21, 45).

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. «Attention is All You Need». In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017 (cit. on p. 23).

[39] Yonghui Wu et al. «Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation». In: 2016. URL: https://arxiv.org/abs/1609.08144 (cit. on p. 23).

[40] Ilya Loshchilov and Frank Hutter. «Decoupled Weight Decay Regularization». In: *International Conference on Learning Representations*. 2019. URL: https://arxiv.org/abs/1711.05101 (cit. on p. 26).

[41] Diederik P Kingma and Jimmy Ba. «Adam: A Method for Stochastic Optimization». In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 26).

[42] Justin Johnson and Taghi M. Khoshgoftaar. «Survey on Deep Learning with Class Imbalance». In: *Journal of Big Data* 6.1 (2019), pp. 1–54 (cit. on pp. 29, 40).

[43] George A Miller. «WordNet: a lexical database for English». In: *Communications of the ACM* 38.11 (1995), pp. 39–41 (cit. on p. 30).

[44] Jia Chen et al. *Data Augmentation for Emotion Detection in Small Imbalanced Text Data*. 2023. URL: https://arxiv.org/abs/2310.17015 (cit. on p. 30).

[45] Andre Goncalves, Partho P Ray, Ben Soper, Rick Stevens, and John Guttag. «Generation and evaluation of synthetic patient data». In: *Nature Machine Intelligence* 2.10 (2020), pp. 593–603 (cit. on p. 30).