PRACTICAL WORK ■ 7–8

Designing the Structure of a Thesis
and the Logic of Scientific Research

Student: Didar Kanalbay | **Group:** SIS-2224

Thesis Topic:

Creating a Multiclass Model for Detecting Destructive Web Content in Social Networks and Instant Messengers Using Machine Learning

2025

1. Purpose of the Practical Work

To develop the ability to design a logically structured thesis that ensures the relationship between the scientific problem, the goal, the objectives, the methods, and the results of the research on the topic of *creating a multiclass model for detecting destructive web content in social networks and instant messengers using machine learning*.

2. Scientific Problem, Goal, and Objectives

2.1 Scientific Problem

Existing content-moderation systems for social networks and instant messengers are predominantly binary (harmful vs. safe) or rely on keyword-based filtering. They fail to distinguish among multiple categories of destructive content (extremism, cyberbullying, self-harm promotion, fraud, disinformation, etc.), leading to high false-positive rates and poor coverage of emerging threats. There is a need for a robust multiclass classification framework that can accurately and efficiently categorize diverse types of destructive web content.

2.2 Goal of the Research

To develop and validate a multiclass machine-learning model capable of detecting and classifying multiple categories of destructive web content in social networks and instant messengers with high accuracy, interpretability, and scalability.

2.3 Research Objectives

Objective 1. To analyze the current state of research on destructive-content detection and identify gaps in multiclass classification approaches.

Objective 2. To define and systematize a taxonomy of destructive web content categories relevant to social networks and instant messengers.

Objective 3. To review and compare machine-learning methods applicable to multiclass text classification (traditional ML, deep learning, transformers).

Objective 4. To collect, preprocess, and annotate a representative multilingual dataset of social-media and messenger content.

Objective 5. To design the architecture of a multiclass classification model and select optimal features and hyperparameters.

Objective 6. To implement the model and conduct experiments evaluating accuracy, precision, recall, F1-score, and inference time.

Objective 7. To compare the proposed model with existing analogues and baseline solutions.

Objective 8. To formulate practical recommendations for integrating the model into content-moderation pipelines.

3. Complete Table of Contents of the Thesis

4. Connection Between Chapters and Research Objectives

The table below maps each research objective to the corresponding thesis section, the methods employed, and the expected results.

#	Research Objective	Thesis Section	Methods	Expected Results
1	Analyze current state of research and identify gaps	1.1, 1.4	Systematic literature review, bibliometric analysis	State-of-the-art overview; identified research gaps
2	Define taxonomy of destructive content categories	1.2	Content analysis, expert consultation, clustering	Comprehensive taxonomy of 6–10 content categories
3	Review and compare ML methods for multiclass classification	1.3, 1.4	Comparative analysis, meta-analysis	Ranked list of applicable methods with pros/cons
4	Collect, preprocess, and annotate dataset	2.3, 2.4	Web scraping, NLP preprocessing, inter-annotator agreement	Labeled multilingual dataset (10k+ samples)
5	Design the multiclass model architecture	2.5, 2.6	Deep learning design, transfer learning, hyperparameter optimization	Model architecture specification and training pipeline
6	Implement model and conduct experiments	3.1–3.4	Cross-validation, ablation study, statistical testing	Performance metrics: Accuracy, macro-F1 ≥ 0.85
7	Compare with existing analogues	3.5	Benchmarking, paired statistical tests	Comparative table showing improvement over baselines
8	Formulate practical recommendations	3.6, Conclusion	Synthesis, case studies	Deployment guidelines and integration recommendations

5. Content Description of Each Section

Introduction

- Relevance of the research: growth of destructive content online, limitations of current moderation.
- Scientific problem, goal, and objectives of the research.
- Object of research: destructive web content in social networks and instant messengers.
- Subject of research: multiclass machine-learning classification of destructive content.
- Methods of research (overview).
- Scientific novelty and practical significance.
- Structure of the thesis.

Chapter 1. Theoretical Foundations of Destructive Web Content Detection

1.1 Analysis of the Subject Area

- Definition of destructive web content and its manifestations (extremism, hate speech, cyberbullying, self-harm, fraud, disinformation).
- Statistics on the prevalence of destructive content in popular social networks (VK, Telegram, Instagram, TikTok) and messengers (WhatsApp, Telegram).
- Legal and ethical frameworks for content moderation.
- Key stakeholders: platforms, regulators, users, researchers.

1.2 Taxonomy and Classification of Destructive Web Content

- Existing classification schemes in the literature.
- Proposed taxonomy: categories (extremism, cyberbullying, self-harm promotion, fraud/phishing, disinformation, drug propaganda, explicit violence) with definitions and boundary criteria.
- Discussion of overlapping and ambiguous categories; multi-label vs. multiclass distinction.

1.3 Review of ML Methods for Multiclass Text Classification

- Traditional methods: SVM, Naive Bayes, Random Forest, Gradient Boosting with TF-IDF/BoW features.
- Deep-learning methods: CNN-text, BiLSTM, attention mechanisms.
- Transformer-based models: BERT, RoBERTa, XLM-R for multilingual scenarios.
- Ensemble and hybrid approaches.
- Evaluation metrics for multiclass problems: macro/micro/weighted F1, Cohen's kappa, confusion matrix analysis.

1.4 Comparative Analysis of Existing Solutions

- Review of commercial systems: Perspective API, Meta's content moderation, OpenAI moderation endpoint.
- Academic systems and benchmark datasets (HateXplain, Davidson et al., Jigsaw Toxic Comment).
- Comparison by: number of classes, languages supported, F1-score, latency, interpretability.

- Identified gaps and justification for the proposed approach.

1.5 Conclusions for Chapter 1

- Summary of theoretical findings; justification for the research direction.

Chapter 2. Methodology and Development of the Multiclass Classification Model

2.1 Formal Problem Statement

- Mathematical formulation: given input text x, predict label y ∈ {c■, c■, …, c■} where k is the number of destructive-content categories + 1 (safe).
- Optimization objective: minimize cross-entropy loss over the training set.
- Constraints: inference time ≤ 100 ms per sample; model size suitable for deployment.

2.2 Selection of Methods, Tools, and Frameworks

- Programming language: Python 3.10+.
- Key libraries: PyTorch / HuggingFace Transformers, scikit-learn, pandas, NLTK/spaCy.
- Experiment tracking: MLflow / Weights & Biases.
- Hardware: GPU cluster (NVIDIA A100 or equivalent).
- Justification for each choice.

2.3 Data Collection, Preprocessing, and Annotation

- Data sources: public datasets (Jigsaw, HateXplain), social-media API crawling (Telegram, VK), web scraping.
- Preprocessing pipeline: text cleaning, normalization, language detection, deduplication.
- Annotation protocol: guidelines, annotator training, inter-annotator agreement (κ ≥ 0.75).
- Data augmentation strategies: back-translation, synonym replacement, paraphrasing.
- Handling class imbalance: oversampling (SMOTE-text), class weights, focal loss.

2.4 Feature Engineering and Text Representation

- Baseline features: TF-IDF n-grams, linguistic features (POS, sentiment, readability).
- Embedding-based: Word2Vec, FastText, contextual embeddings from BERT/XLM-R.
- Multimodal signals (optional): metadata features (posting time, user history, URL presence).

2.5 Architecture Design of the Multiclass Classification Model

- Base model: fine-tuned XLM-RoBERTa-base (multilingual support).
- Classification head: linear layer with softmax for k classes.
- Alternative architectures explored: BiLSTM + Attention, CNN-text ensemble.
- Architecture diagram and detailed layer description.

2.6 Training Strategy and Hyperparameter Tuning

- Training procedure: learning rate scheduling (warm-up + cosine decay), early stopping.
- Hyperparameter search: Bayesian optimization over learning rate, batch size, dropout, weight decay.

- Regularization: dropout, label smoothing, gradient clipping.
- Cross-validation scheme: stratified 5-fold.

2.7 System Architecture for Deployment

- End-to-end pipeline: data ingestion → preprocessing → inference → result storage.
- REST API design (FastAPI), containerization (Docker), orchestration considerations.
- Monitoring and retraining pipeline.

2.8 Conclusions for Chapter 2

- Summary of methodological decisions and their justification.

Chapter 3. Experimental Evaluation and Results

3.1 Description of the Dataset and Experimental Setup

- Final dataset statistics: total samples, class distribution, language distribution.
- Train/validation/test split ratios (70/15/15) with stratification.
- Hardware and software environment for experiments.

3.2 Experimental Methodology and Evaluation Metrics

- Primary metrics: macro-F1, weighted-F1, accuracy.
- Secondary metrics: per-class precision, recall, confusion matrix, ROC-AUC (one-vs-rest).
- Statistical significance: paired bootstrap test, confidence intervals.
- Ablation study design: effect of preprocessing steps, features, model components.

3.3 Results of Multiclass Classification Experiments

- Performance of baseline models (SVM, Logistic Regression, Random Forest).
- Performance of deep-learning models (BiLSTM, CNN-text).
- Performance of transformer models (BERT, XLM-RoBERTa).
- Summary results table with all metrics.

3.4 Analysis and Interpretation of Results

- Class-level analysis: which categories are hardest to classify and why.
- Feature importance and attention visualization.
- Error patterns: common misclassifications, edge cases.

3.5 Comparison with Existing Analogues and Baselines

- Benchmark comparison on public datasets (Jigsaw, HateXplain).
- Comparison with commercial APIs (Perspective API).
- Improvement quantification: absolute and relative gains in F1.

3.6 Error Analysis and Model Interpretability

- LIME / SHAP explanations for sample predictions.
- Adversarial robustness testing (typos, obfuscation, code-switching).
- Recommendations for model improvement.

3.7 Conclusions for Chapter 3

- Summary of experimental findings and their implications.

Conclusion

- Summary of research contributions.
- Degree of achievement of each objective.
- Scientific novelty and practical significance.
- Limitations of the study.
- Directions for future research.

References

- Minimum 40–60 sources: journal articles, conference proceedings, standards, online resources.

Appendices

- Appendix A: Full dataset statistics and sample annotations.
- Appendix B: Source code of key modules.
- Appendix C: Additional experimental results and confusion matrices.
- Appendix D: Glossary of terms.

6. Logical Model of the Thesis Structure

The logical model below shows the chain of reasoning that connects the scientific problem to the final conclusions of the thesis.

Scientific Problem	Lack of effective multiclass classification for destructive web content in social networks and messengers
↓	
Goal	Develop and validate a multiclass ML model for detecting diverse types of destructive content
↓	
Objectives (8 items)	O1: Literature review | O2: Taxonomy O3: Method comparison | O4: Dataset O5: Model design | O6: Experiments O7: Benchmarking | O8: Recommendations
↓	
Chapters	Ch 1: Theory (O1–O3) Ch 2: Methodology (O4–O5) Ch 3: Experiments (O6–O8)
↓	
Methods	Literature review, NLP preprocessing, Deep learning, Transfer learning, Statistical evaluation, Benchmarking
↓	
Results	Taxonomy of destructive content, Annotated dataset, Trained multiclass model, Performance metrics (F1 ≥ 0.85)
↓	
Conclusions	Validated model, Practical guidelines, Future research directions

7. Awareness of Common Mistakes

The following table lists common student mistakes and the specific measures taken in this thesis to avoid them.

Common Mistake	How It Is Avoided in This Thesis
Lack of connection between the structure and research objectives	Every section is explicitly mapped to one or more research objectives (see Section 4). The table of contents was designed objectives-first.
Overly general section titles	Section titles are specific and reflect concrete research activities (e.g., “Taxonomy and Classification of Destructive Web Content Categories” instead of “Literature Review”).
Inconsistency between practical and theoretical parts	Chapter 1 provides the theoretical justification for every method used in Chapters 2–3. The comparative analysis in 1.4 directly motivates the design decisions in 2.5.
Duplication of content	Each section has a clearly defined scope. Conclusions at the end of each chapter summarize without repeating body text. Cross-references are used instead of duplication.

Prepared by: Didar Kanalbay | **Group:** SIS-2224

Practical Work ■ 7–8 — Designing the Structure of a Thesis and the Logic of Scientific Research