Software Requirements Specification

for

<Transformers experiments>

Version 1.0 approved

Prepared by <Andrii Martynenko>

<Kharkiv National University of Radioelectronics>

<June 2025>

Table of Contents

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 1

2.1 Product Perspective 1

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 2

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 3

4.1 Load CSV File 3

4.2 Query Clustreing 3

4.3 Output of results 4

4.4 Interaction with results 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 4

5.3 Security Requirements 4

5.4 Software Quality Attributes 4

5.5 Business Rules 4

6. Other Requirements 5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Initial release | June 2025 | Initial version completed | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

This document defines the functional and non-functional requirements for a BERT-based sentiment classification system that processes the IMDb dataset. The system uses the Transformers library and PyTorch to build, train, and evaluate a binary text classifier.

## Document Conventions

Functional requirements are marked "FR", non-functional ones "NFR". Priority levels are High (H), Medium (M), Low (L).

## Intended Audience and Reading Suggestions

The document is intended for machine learning engineers, researchers, students, and educators. It is recommended to begin with Sections 2 and 4.

## Product Scope

The system is a script-based machine learning tool that performs sentiment classification using BERT. It loads data, preprocesses it, trains the model, and evaluates performance.

## References

* HuggingFace Transformers Documentation
* IMDb Dataset (HuggingFace Datasets Library)
* PyTorch Documentation
* scikit-learn Documentation

# Overall Description

## Product Perspective

This project is a standalone Python-based training pipeline, suitable for academic and research purposes. It can serve as a base for further fine-tuning or adaptation.

## Product Functions

 Load IMDb dataset

 Balance train/validation split

 Tokenize texts using AutoTokenizer

 Train BERT model via HuggingFace Trainer API

 Compute evaluation metrics (precision, recall, F1)

## User Classes and Characteristics

**ML Engineers/Researchers** – basic Python and PyTorch knowledge required

## Operating Environment

 OS: Windows/Linux/MacOS

 Python 3.10+

 CUDA-enabled GPU (optional)

## Design and Implementation Constraints

 Pretrained model: bert-base-uncased

 Max token length: 128

 Libraries: transformers, datasets, scikit-learn, pandas, numpy

## User Documentation

 Jupyter Notebook or inline script comments

 Optional PDF guide

## Assumptions and Dependencies

 IMDb dataset is available via HuggingFace

 Internet access is required for downloading pretrained models

# External Interface Requirements

## User Interfaces

 Command-line execution

 Logs printed to console during training

## Hardware Interfaces

Optional GPU usage for faster training

## Software Interfaces

 Transformers

 Datasets

 PyTorch

 scikit-learn

 pandas, numpy

## Communications Interfaces

Not applicable (no network communication involved)

# System Features

## Load Dataset

REQ-1: The system shall load the IMDb dataset using datasets.load\_dataset().

REQ-2: The system shall convert the dataset to pandas.DataFrame.

## Tokenization and Preprocessing

REQ-3: The system shall tokenize input using AutoTokenizer.

REQ-4: The tokenizer shall apply padding and truncation to max length 128.

REQ-5: The system shall convert tokenized data into PyTorch tensors.

## Model Training

REQ-6: The system shall initialize training using Trainer.

REQ-7: The system shall train for 2 epochs.

REQ-8: The training process shall log to the ./logs directory.

REQ-9: The system shall evaluate the model at each epoch.

## Evaluation

REQ-10: The system shall compute precision, recall, and F1 (macro).

REQ-11: The metrics shall be calculated using precision\_recall\_fscore\_support.

# Other Nonfunctional Requirements

## Performance Requirements

NFR-1: The model should complete training on 20,000 samples within 1 hour on a standard GPU.

## Safety Requirements

NFR-2: The system shall validate that both text and label fields exist before training.

## Security Requirements

NFR-3: Not applicable, as the system does not process personal or sensitive data.

## Software Quality Attributes

* Usability: clean and intuitive interface
* Reliability: file format validation
* Portability: .exe launch capability

## Business Rules

 No personal data used

 Only binary sentiment classification (positive/negative)

# Other Requirements

 Optional support for other transformer models

 Future adaptation for multi-class classification

 Possible GUI wrapper for user-friendliness