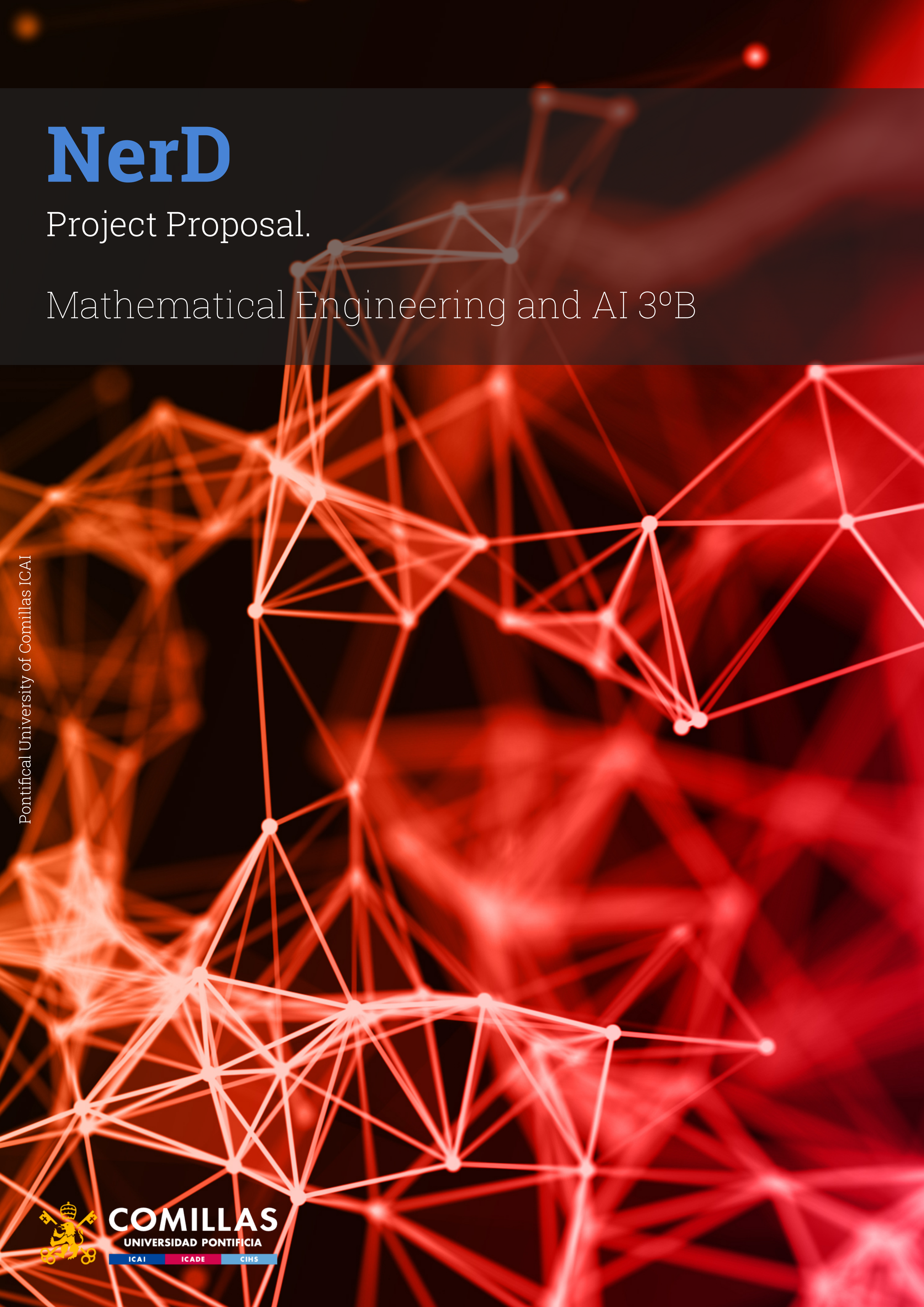

NerD

Project Proposal.

Mathematical Engineering and AI 3ºB

Pontifical University of Comillas ICAI

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

NerD

Project Proposal.

Mathematical Engineering and AI 3ºB

carried out by

Jorge Kindelán Navarro
Eugenio Ribón Novoa
Beltrán Sánchez Careaga
Ignacio Queipo de Llano Pérez-Gascón

Professors:	Jaime Pizarroso Gonzalo Andrés Occhipinti Liberman
Project Duration:	March 2025 - April 2025
Faculty:	Pontifical University of Comillas

Contents

1 Introduction

This project aims to develop an automatic alert generation system that integrates Named Entity Recognition (NER) and Sentiment Analysis (SA). By analyzing textual data from news articles and social media, the system will identify key entities, assess sentiment, and generate meaningful alerts.

The entire process, including data preprocessing, model training, and final evaluations, will be documented and reflected in the NerD repository on GitHub: NerD Repository

1.1. Related Work

1.1.1. Named Entity Recognition (NER)

Previous research has demonstrated the effectiveness of LSTM-based architectures for entity recognition. Benchmark datasets such as CoNLL-2003 and OntoNotes 5.0 are commonly used for training and evaluation. LSTM-based models have shown great promise in extracting named entities from unstructured text, and the performance of these models can be further enhanced with pre-trained embeddings or transfer learning.

For more details, refer to the following: - CoNLL-2003 Dataset - OntoNotes 5.0

1.1.2. Sentiment Analysis (SA)

Datasets like Sentiment140 and Financial PhraseBank are widely used for sentiment classification. Traditional models leverage LSTMs and GRUs, while more recent methods use Transformers (which are not allowed for SA and NER in this project). Sentiment analysis has become a key task in natural language processing, with applications ranging from social media monitoring to market analysis.

For more information on sentiment analysis, check out: - Sentiment140 Dataset - Financial PhraseBank

1.1.3. Multitask Learning for NER and SA

Some studies have explored joint architectures that optimize shared representations for both tasks, improving overall performance. By training models on both Named Entity Recognition and Sentiment Analysis simultaneously, multitask learning can help improve the generalization ability of the model, leading to more robust and accurate predictions. Recent advancements in multitask learning techniques, such as joint attention mechanisms, further enhance the performance of these systems.

For further reading on multitask learning for NER and SA, refer to: - Multitask Learning for Named Entity Recognition and Sentiment Analysis

2
Datasets

To train a model capable of handling both Named Entity Recognition (NER) and Sentiment Analysis (SA), we propose using a combination of specialized datasets. The selected datasets will be carefully preprocessed and adapted to fit our training pipeline, ensuring that both the NER and SA tasks are well-supported. Below are the datasets we intend to use:

2.1. MultiNERD

The **MultiNERD** dataset provides labeled data specifically for Named Entity Recognition (NER). It includes a wide variety of named entities, such as persons, locations, organizations, dates, and more, making it a valuable resource for training models to extract meaningful entities from text. MultiNERD is an open-source dataset that supports multiple languages and can be applied in diverse domains. More information, including access to the data, can be found on the project's GitHub repository: GitHub Link.

2.2. Sentiment Labels

While **MultiNERD** provides excellent data for NER, there is currently no dataset that pairs sentiment annotations with it. To address this gap, we propose using a high-accuracy pre-trained sentiment analysis (SA) model to generate sentiment labels for the text data.

This approach will enable us to augment the MultiNERD dataset with sentiment information, ensuring that our model can simultaneously perform both NER and SA.

3

Model Architecture

Our system integrates pre-trained models with custom-trained components:

3.1. 1. Image Processing Module (Pretrained Model)

The first module in our architecture focuses on processing images to generate textual descriptions. This task is achieved by utilizing a pre-trained image captioning model. For this purpose, models such as **CLIP** (Contrastive Language-Image Pretraining) can be employed.

3.2. 2. Text Processing and Feature Extraction

In the second module, we focus on processing the textual input data. The text will undergo the following steps:

- **Text Embedding**: The input text is embedded using a same pre-trained embedding model (e.g., **Word2Vec**, **GloVe**, or **FastText**).
- **Unified Representation**: Once embeddings for both the input text and image captions are generated, they are concatenated into a single vector, providing a richer input for the next module.

3.3. 3. Custom NER and SA Model (Trained from Scratch)

The third module involves a custom multitask model designed to handle both Named Entity Recognition (NER) and Sentiment Analysis (SA). This model is trained from scratch using the unified input representation (from both text and image caption embeddings).

The model architecture includes the following components:

- **BiLSTM Layer**: The combined input is passed through a Bidirectional Long Short-Term Memory (BiLSTM) layer. The BiLSTM captures sequential dependencies from both directions (forward and backward) in the text, allowing the model to understand the context better and capture long-range dependencies.
- **NER Head**: The first output head is responsible for extracting named entities from the input. These entities could include person names, organization names, locations, dates, and more. The NER head will output labels corresponding to the identified entities.
- **SA Head**: The second output head classifies the sentiment of the input text, determining whether the sentiment is positive, negative, or neutral. This head outputs a sentiment label based on the content of the input.

3.3.1. Alert Generation Module

The alert generation module operates based on predefined rules or thresholds, ensuring that important insights or anomalies do not go unnoticed, helping users take immediate action when necessary.

4

Project Plan and Milestones

The project will be carried out over the course of four weeks, with each week dedicated to specific tasks and milestones to ensure smooth progression. Below is a breakdown of the plan:

4.1. Week 1: Data Collection and Preprocessing

- Identify and preprocess datasets containing both NER and SA annotations.
- Augment data using Large Language Models (LLMs) to generate additional labeled samples if necessary.

4.2. Week 2-3: Model Development

- Implement separate LSTM-based models for NER and SA.
- Experiment with joint architectures optimizing for both tasks simultaneously.

4.3. Week 3: Alert Generation Module

- Develop rule-based and neural-based approaches for alert generation.
- Integrate outputs from NER and SA models into a structured alert system.

4.4. Week 4: Evaluation and Refinement

- Measure performance using F1-score for NER and accuracy for SA.
- Optimize hyperparameters and adjust data preprocessing strategies.

4.5. Week 4: Final Integration and Report Writing

- Compile results, generate final system outputs, and document findings.
- Prepare the final project deliverables.

This proposal outlines a structured approach to developing an advanced multitask learning system for NER and SA, with a strong focus on practical applications in alert generation. Since this is a preliminary proposal, modifications and adjustments may be required as the project progresses.