UNIVERSITY OF BUEA

P.O. Box 63,

Buea, South West Region

CAMEROON

Tel: (237) 3332 21 34/3332 26 90

Fax: (237) 3332 22 72

REPUBLIC OF CAMEROON

PEACE-WORK-FATHERLAND

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

NAMED ENTITY RECOGNITION FOR UNDER-RESOURCED LANGUAGES: CASE STUDY OF YEMBA LANGUAGE

A dissertation submitted to the Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea, in Partial Fulfilment of the Requirements for the Award of Bachelor of Engineering (B.Eng.) Degree in Computer Engineering

By:

DJIOTSA DJOUAKE CHRISTIAN DARYN

Matriculation Number: FE20A029

Option: Sotfware Engineering

Supervor:

Dr. Sop Deffo

University Of Buea

2023/2024 Academic Year

NAMED ENTITY RECOGNITION FOR UNDER-RESOURCED LANGUAGES: CASE STUDY OF YEMBA LANGUAGE

DJIOTSA DJOUAKE CHRISTIAN DARYN

Matriculation Number: FE20A029

2023/2024 Academic Year

Dissertation submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (B.Eng.) Degree in Computer Engineering.

Department of Computer Engineering

Faculty of Engineering and Technology

University of Buea

CERTIFICATION OF ORIGINALITY

We the undersigned, herby certify that this dissertation titled **"NAMED ENTITY RECOGNITION FOR UNDER-RESOURCED LANGUAGES: CASE STUDY OF YEMBA LANGUAGE"**, presented by Djiotsa Djouake Christian Daryn, **Matriculation number FE20A029** has been carried out his in the Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea under the supervision of **Dr Sop Deffo**.

This dissertation is authentic and represents the fruit of his research and efforts.

Date_____

Student   **Supervision**

_____   _____

Head of Department

_____

DEDICATION

To my lovely mother **Mrs. FOFE EDWIGE.**

ACKNOWLEDGMENT

First of all, I would like to express my gratitude to the Faculty Of Engineering And Technology of University Of Buea for giving me this opportunity to go out and experience the real enterprise engineering world. I also thank the Dean of the Faculty of Engineering and Technology Prof. Agbor Dieudonne Agnor whose under his supervision I was equipped with adequate knowledge on computer engineering to carry out this project. I express my gratitude to my academic supervisor Dr. Sop Deffo for everything he did for me to get this project functional. And finally, I express my gratitude to family and all my friends for the support and any other person who has contributed directly or indirectly to the production of this report.

ABSTRACT

This study addresses the digital gap for the Yemba language (Cameroon, 300k+ speakers) by developing a Named Entity Recognition (NER) system. NER identifies entities like people and locations in text. The system leverages pre-trained transformer models, fine-tuned for Yemba using limited annotated data. This enables effective entity recognition, promoting Yemba cultural heritage preservation, information access, and community empowerment in the digital age. This paves the way for future NLP advancements in under-resourced languages.

Table of Contents

vii

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
CPU	Central Processing Unit
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
FET	Faculty Of Engineering And Technology
MVT	Model View Template
NER	Named Entity Recognition
NLP	Natural Language Processing
RESTFUL	Representational State Transfer
XLM-RoBERTa	Cross-lingual Language Model Roberta
YembaNER	Yemba Named Entity Recognition

CHAPTER 1: GENERAL INTRODUCTION

1.1 Background and Context of Study

According to Statista (a German online platform that specializes in data gathering and visualization) report in 2022, there are more than 2000 languages in Africa. Across Africa, a staggering variety of languages paint a rich picture of the continent's cultural heritage. This diversity, however, faces a threat. The Oxford Press reports that over 308 languages are critically endangered. Sadly, around 100 of these languages are concentrated in just four countries: Cameroon, Chad, Ethiopia, and Nigeria. Furthermore, UNESCO reports that more than 70% of Cameroon languages are endangered languages. In addition, the majority of languages spoken in Cameroon such as Afade, Aghem, Akum, Yemba, Ambele and so on are under-resourced.

The **Yemba language**, spoken by over 300,000 people in Cameroon's Western Province(Lebialem division, Menoua division and Dschang area) since 1992 according to SIL Cameroon, is a vibrant thread in the rich tapestry of Bamiléké languages. It embodies a unique cultural heritage and complex linguistic features. However, Yemba's status as an under-resourced language creates a digital divide, hindering efforts to:

- **Preserve Cultural Heritage**: The scarcity of Yemba text data online poses a threat to the preservation and dissemination of Yemba cultural knowledge for future generations.
- **Bridge the Language Barrier:** Search engines and online tools often lack the ability to understand Yemba, making it difficult for speakers to access information readily available in dominant languages.

- **E**mpower **Yemba Communities:** The lack of Yemba language resources limits educational opportunities and hinders technological advancements within Yemba communities.

1.2. Problem Statement

The Yemba language, a vibrant thread in the tapestry of Cameroon's Bamiléké languages, faces significant obstacles in the digital age. Despite its rich cultural heritage and complex linguistic features, spoken by over 300,000 people in the Western Province, Yemba's status as an under-resourced language creates a digital divide that hinders progress on multiple fronts. Amount which:

- **L**imited **Digital Corpus**: The scarcity of Yemba text data online impedes efforts to preserve cultural heritage and disseminate knowledge for future generations. The lack of a substantial digital corpus creates a barrier to effective language documentation and analysis.
- **Language Barrier in the Digital Sphere:** Search engines and online tools are often unable to process Yemba text, making it difficult for speakers to access information readily available in dominant languages. This disparity limits Yemba speakers' ability to participate fully in the digital world.
- **Educational Disparity:** The lack of Yemba language resources restricts educational opportunities and technological advancements within Yemba communities. The absence of tools tailored to the Yemba language hinders progress in education and literacy initiatives.

Developing a Yemba Named Entity Recognition (NER) system specifically designed for the Yemba language is a crucial step towards overcoming these challenges.

1.3.   Objectives of the Study

1.3.1   General Objective

Named Entity Recognition (NER) is a sub-task of Natural Language Processing (NLP) that aims to identify and classify named entities within text data. These entities can be people, organizations, locations, dates, monetary values, etc. NER plays a crucial role in various NLP applications, including information extraction, machine translation, and question answering.

Unfortunately, developing NER models for low-resource languages like Yemba presents significant challenges due to the limited availability of annotated data and language-specific resources. This report explores the development of a Yemba NER system, addressing the challenges of under-resourced languages.

The main goal of the Yemba Named Entity Recognition is to develop a Named Entity Recognition (NER) system that accurately identifies and classifies named entities within Yemba text data.

1.3.2.   Specific Objectives

Building upon the main objective of the Yemba NER, here are the specific objectives that will be further explored:

- **Entity Type Coverage:** Define a comprehensive set of relevant named entity types for the Yemba language.

- **Data Acquisition and Annotation Strategy:** Develop a plan to address the data scarcity challenge.

- **Model Selection and Adaptation:** Choose a suitable NER model architecture that can perform effectively with limited Yemba data. This involve:

  - Investigating pre-trained NER models on large, general-domain datasets and adapting them to Yemba through transfer learning or fine-tuning techniques.

  - Exploring architectures specifically designed for low-resource language settings, known for requiring less data for training.

- **Evaluation Strategy:** Define clear metrics to evaluate the performance of the Yemba NER model. Standard metrics like precision, recall, and F1-score can be used, potentially focusing on specific entity types if relevant.

- **Performance Benchmark:** Establish a target performance benchmark for the Yemba NER model. This benchmark could be based on existing NER models for other low-resource languages or a specific level of accuracy required for the intended NLP application.

- **Web Application Development:** Design and develop a user-friendly web application that provides access to the Yemba NER system. This application should allow users to:

  - Upload or paste Yemba text data for NER processing.

  - Visualize the identified named entities within the text.

1.4. Proposed Methodology

The development of a Yemba NER System that accurately identifies and classifies named entities in Yemba text data, requires a good methodology. This methodology focuses on a promising approach for developing a Yemba NER system by leveraging the power of pre-trained

4

transformers and fine-tuning techniques. By carefully addressing data challenges and considering the specific needs of the Yemba language.

Following is the step by step process choose to build the Yemba name entity recognition system:

i. Data Acquisition and Preprocessing:

- Gather Yemba text data for training and testing the NER model. This could involve:
  - Collaborating with Yemba speakers to create a corpus of annotated text data.
- Preprocess the text data by:
  - Cleaning and normalizing the text (e.g., managing punctuation, converting to lowercase or uppercase depending on the model's requirements).
  - Tokenizing the text into words: Concis of splitting the text into individual words.

ii. Entity Definition and Annotation:

(1) **Define the specific named entity types you want the model to recognize in Yemba text.** This could include:
  - ✓ People (PER)
  - ✓ Geographic Locations (GEO): For cities, location, any place but not geopolitical location
  - ✓ Time (TIME): For date, days of the week, and month of year
  - ✓ Quantity (QUAN): For measurement unit like, money, percentage and distance

(2) **Annotate the preprocessed Yemba text data according to the defined entity types. which was done manually**.

iii. Model Selection and Fine-tuning: Utilize the pre-trained BERT-base-multilingual-cased model as the foundation for your NER system. And fine-tune for our label dataset.

iv. Training and Evaluation:

- Split the annotated Yemba data into training, validation, and test sets.

- Train the fine-tuned BERT model on the training set, monitoring its performance on the validation set to prevent overfitting.

- Evaluate the final model's performance on the held-out test set using standard NER metrics like precision, recall, and F1-score for each defined entity type.

v. Model Deployment and Integration:

- Deploy the fine-tuned BERT model for Yemba NER into a web or API. This would allow users to submit Yemba text data and receive the identified named entities.

1.5. Significance of the Study

Named Entity Recognition (NER) is a crucial component of Natural Language Processing (NLP) tasks, enabling the identification and classification of named entities like people, locations, and organizations within text. This study on NER for under-resourced languages(case study of Yemba language) holds significance for several reasons.

The development of an NER system for under-resourced languages holds significant value on multiple levels:

- **Bridging the Resource Gap:** Under-resourced languages lack readily available NLP tools. This study addresses this gap by creating a dedicated NER system, paving the way for future NLP applications for these languages.

- **Boosting NLP Capabilities:** A functional NER system empowers the development of advanced NLP applications like:

  - **Machine Translation:** Enabling efficient translation between Yemba and other languages.

  - **Information Retrieval:** Facilitating the search and retrieval of relevant Yemba text data.

  - **Sentiment Analysis:** Extracting sentiment from Yemba text for various applications.

- **Cultural Heritage Preservation:** NER can be instrumental in processing and analyzing Yemba historical documents and cultural materials, aiding in their preservation and accessibility.

- **Endangered Language Support:** This study serves as a valuable model for developing NER systems for other endangered or under-resourced languages, promoting their computational support.

1.6. Scope of the Study

This study focuses on developing a Named Entity Recognition system for the Yemba language. It will explore:

- **Entity Type Definition:** Defining the specific entity types relevant to Yemba, such as:
  - **People:** Names of individuals, including titles and honorifics.
  - **Locations:** Countries, cities, villages and continents.
  - **Quantity:** Distance, money, and percentages.

- **Time:** Dates, days of the weeks and months of the year.

- **Data Acquisition and Preprocessing:**

  - Strategies for acquiring a suitable corpus of Yemba text data.

  - Techniques for cleaning, annotating, and preparing the data for model training and evaluation will be addressed.

- **Model Selection and Training:**

  - The study will investigate different NER model architectures suitable for the Yemba language and available computational resources.

  - Training methodologies, including hyperparameter tuning, will be explored to optimize the model's performance.

- **Evaluation Methods:**

  Establishing effective evaluation metrics to assess the performance of the NER system is crucial. This might include: Precision, Recall, and F1-score for the training and provide an accuracy for each defined entity.

1.7. Delimitation of the Study

The study will focus on the core functionalities of NER for Yemba. The following aspects will be outside the scope:

- **Deep Linguistic Analysis:** A comprehensive analysis of Yemba grammar and morphology might be beyond the immediate focus. However, a basic understanding of Yemba language structure is essential for effective NER model development.

- **Real-World Applications:** While the study paves the way for NLP applications, integrating the NER system into specific real-world tasks might be explored in future work.

- **Multilingual NER:** The initial focus will be on Yemba as a single language. However, the potential for extending the system to handle multilingual NER tasks can be explored further.

1.8. Definition of Keywords and Terms

(1)**Named Entity Recognition (NER):** The process of automatically identifying and classifying named entities within text data, such as persons, locations, and organizations. This task is fundamental in many NLP applications, providing structured information from unstructured text.

(2)**Under-resourced Language:** A language with limited computational resources and tools available, which hinders the development of NLP applications. These languages often lack extensive annotated datasets, linguistic resources, and pre-trained models, making NLP research and development more challenging.

(3)**Natural Language Processing (NLP):** A field of computer science concerned with the interaction between computers and human language. NLP enables machines to understand, interpret, and generate human language, facilitating a wide range of applications from machine translation to sentiment analysis.

(4)**Fine-tuning:** The process of adapting a pre-trained model for a specific task by adjusting its final layers. This allows the model to learn task-specific patterns, leveraging the general knowledge acquired during pre-training to improve performance on the target task.

(5)**Evaluation Metrics:** Quantitative measures used to assess the performance of a machine learning model on a specific task. For NER, common evaluation metrics include Precision, Recall, and F1-score, which provide insights into the accuracy and reliability of the entity recognition process.

(6)**Transformer:** A type of deep learning model architecture that uses self-attention mechanisms to process and generate sequences of data. Transformers have revolutionized NLP by enabling the development of powerful models like BERT and GPT, which achieve state-of-the-art performance on various NLP tasks.

(7)**Backend:** The part of a software system responsible for processing business logic, database interactions, authentication, and other server-side functions. The backend supports the frontend by handling requests, performing operations, and returning responses.

(8)**Frontend:** The part of a software system that users interact with directly. It includes the user interface (UI) and client-side logic, typically implemented using web technologies like HTML, CSS, and JavaScript frameworks.

(9)**Transfer Learning:** A machine learning technique where a model developed for one task is reused as the starting point for a model on a second task. This is particularly useful in NLP for adapting pre-trained models to new tasks or languages with limited data.

(10)   **Precision Evaluation Metric:** A measure of a model's accuracy in identifying relevant instances among the retrieved instances. In NER, precision is the ratio of correctly identified entities to the total number of entities identified.

(11)   **Recall Evaluation Metric:** A measure of a model's ability to identify all relevant instances in the dataset. In NER, recall is the ratio of correctly identified entities to the total number of actual entities in the text.

(12)   **F1 Score:** A metric that combines precision and recall to provide a single performance score. It is the harmonic mean of precision and recall, offering a balanced measure of a model's accuracy and completeness.

(13)   **RESTful API:** An application programming interface (API) that adheres to the principles of Representational State Transfer (REST). RESTful APIs use standard HTTP methods and are designed to be stateless, scalable, and easy to use for communication between client and server.

(14)   **Docker:** A platform that allows developers to automate the deployment of applications inside lightweight, portable containers. Containers package an application and its dependencies, ensuring consistency across different environments.

(15)   **Deployment:** The process of making a software application available for use. This involves moving the application from a development environment to a production environment, where it can be accessed by users.

(16)   **Hosting:** The service of providing the infrastructure and resources necessary to run applications or websites on the internet. Hosting ensures that applications are accessible to users by storing them on servers and providing network connectivity.

1.9. Organization of the Dissertation

(1)**Title page**: This page includes the school logo, the title of the project, the author name, the name of the academic institution, the name of the academic supervisor and the date of submission.

(2)**Abstract:** This is a brief summary of this topic that provides an overview of the research question, methods, and main findings.

(3)**Table of contents:** This page lists the chapter headings and subheadings, as well as the page numbers where they can be found.

(4)**Introduction:** This chapter introduces the research question, provides background information on the topic, and outlines the scope and purpose of the dissertation.

(5)**Literature review:** This chapter provides a critical review of the existing system on the research topics, highlighting the gaps and limitations in the current knowledge.

(6)**Analysis and Design:** This chapter describes the various stages of the design of this solution then describes in detail each phase in the resolution process. It also describes the research methods that were used to collect and analyze my data, including tools that were used.

(7) **Implementation/Results:** This chapter presents the findings of my research, how to run the project, Description of the implementation process and the tools and libraries I used.

(8) **Conclusion:** This chapter summarizes the research, highlights the key findings, and provides recommendations for future research.

(9) **References:** This section lists all of the sources that was cited in this dissertation

(10) **Appendices:** This section includes additional materials that are relevant to this research.

II. CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

The development of Named Entity Recognition (NER) systems has been a focal point in the field of Natural Language Processing (NLP) due to its importance in structuring information from unstructured text. NER systems automatically identify and classify named entities such as persons, locations, and organizations within text data. While there has been significant progress in NER for resource-rich languages, many under-resourced languages, like Yemba, remain underexplored. These languages, characterized by limited computational resources and tools, present unique challenges for NLP development.

Recent advancements in machine learning, particularly transfer learning and fine-tuning, have opened new avenues for addressing the limitations faced by under-resourced languages. Transfer learning leverages pre-trained models to enhance performance on new tasks with limited data, showing promise in improving NER capabilities across various languages. Evaluation metrics such as precision, recall, and F1 score are crucial for assessing the effectiveness of these models, providing insights into their accuracy and completeness.

The backend and frontend aspects of system development are also essential in deploying and ensuring the usability of NER applications. RESTful APIs facilitate communication between client-side and server-side components, while tools like Docker ensure consistent deployment across different environments. Hosting services further ensure that applications are accessible and operational.

This chapter explores the existing research and methodologies applied in NER, focusing on the unique challenges and solutions pertinent to under-resourced languages like Yemba.

2.2. General Concepts on Named Entity Recognition (NER)

2.2.1. Named Entity Recognition (NER)

NER is a crucial component of Natural Language Processing (NLP) tasks. It automates the identification and classification of named entities within text data. These entities can be people, locations, organizations, dates, monetary values, or other relevant categories depending on the specific task. NER plays a vital role in various NLP applications, such as:

- **Machine Translation:** NER facilitates accurate translation by identifying and handling named entities appropriately.
- **Information Retrieval:** By recognizing named entities, information retrieval systems can improve the search process and deliver more relevant results.
- **Sentiment Analysis:** NER can aid in sentiment analysis by identifying entities related to the subject of the sentiment.
- **Question Answering Systems:** NER empowers question answering systems to understand and respond to questions that involve named entities.

2.2.2. Challenges of Under-Resource Languages

While NER has become a well-established field, significant challenges arise when dealing with under-resourced languages. These languages are characterized by:

- **Limited Computational Resources:** A scarcity of readily available NLP tools and libraries specifically designed for the language.

- **Limited Training Data:** The lack of large, annotated text corpora hinders the development of robust NER models.

- **Language-Specific Complexities:** Under-resourced languages might have unique grammatical structures, morphology, or naming conventions that require specialized approaches for NER.

2.3. Related Works

Named Entity Recognition (NER) is a crucial task in natural language processing (NLP) that involves identifying and classifying proper names in text into predefined categories such as persons, organizations, locations, and others. While significant progress has been made in developing NER systems for high-resource languages like English, the same cannot be said for under-resourced languages. These languages often lack extensive annotated corpora and computational resources, making the development of effective NER systems particularly challenging.

Named Entity Recognition (NER) for under-resourced languages, exemplified by functional Swahili(spoken in Tanzania, Kenya, Uganda, Rwanda, Burundi, Democratic Republic of the Congo, Somalia, Comoros and Mozambique) NER system developed using annotated dataset and transfer learning, faces significant challenges due to limited resources, but projects like "MasakhaNER" demonstrate that creating high-quality datasets, leveraging advanced machine learning techniques, and involving local communities can effectively extend NER capabilities to diverse languages, contributing to linguistic preservation and enhancing the inclusivity of NLP technologies globally.

2.3.1. Masakha NER

The MasakhaNER project is an innovative effort to improve Named Entity Recognition (NER) for ten African languages that are often overlooked in natural language processing (NLP) research. The project focuses on creating high-quality datasets for these languages, which are necessary for training and testing NER models.

Methodology:

- **Supervised Learning**: Models are trained using annotated data to help them recognize and classify entities.
- **Transfer Learning:** Models trained on well-resourced languages are adapted to work with African languages, using the existing knowledge to improve their performance.
- **Multilingual Models:** Models like mBERT and XLM-R, which are trained in multiple languages, are fine-tuned to handle NER tasks for these African languages.

Evaluation of the project:

The project has tested these methods thoroughly and found them effective in improving NER for these languages. Native speakers and language experts help ensure the datasets are accurate and culturally relevant.

Weaknesses of the project:

The MasakhaNER project encounters several challenges, including limited annotated data affecting model performance and transfer learning issues that may not fully address the specifics of African languages. Multilingual models like mBERT and XLM-R can be inconsistent in performance and may struggle with language nuances.

Significance of the project:

16

The "MasakhaNER" project has a significant impact by promoting linguistic diversity and making NLP technologies more inclusive. It also sets a useful example for similar projects in other under-resourced languages, helping to create a more inclusive global NLP landscape.

2.3.2. Swahili NER

The Swahili NER project is a specialized model designed to enhance Named Entity Recognition (NER) for the Swahili language. This model is part of the broader effort to improve NLP tools for under-resourced languages by leveraging state-of-the-art techniques in machine learning.

Methodology:

- **Supervised Learning:** The model is fine-tuned on a dataset annotated with Swahili entity labels. This training helps the model recognize and classify entities such as names, locations, and organizations within Swahili text.

- **Transfer Learning:** By starting with the pre-trained XLM-RoBERTa model, which has been trained on multiple languages, the model benefits from existing linguistic knowledge. This transfer learning approach enhances its ability to perform NER for Swahili despite the relatively limited amount of Swahili-specific training data.

- **Multilingual Models:** XLM-RoBERTa is a multilingual model that has been trained on a wide range of languages. It is fine-tuned specifically for Swahili, allowing it to adapt its broad linguistic capabilities to the nuances of Swahili NER tasks.

Evaluation of the Model:

The fine-tuned XLM-RoBERTa model has been tested for its effectiveness in recognizing and categorizing entities in Swahili text. The evaluation process involves checking the model's performance on a separate test dataset to ensure accuracy. The model has shown promising results in identifying entities correctly.

Weaknesses Identified:

- **Limited Training Data:** Despite the fine-tuning, the model's performance can be constrained by the amount of high-quality annotated Swahili data available. More extensive and diverse datasets could further improve accuracy.

- **Language Nuances:** Swahili, like many languages, has unique linguistic features and variations. The model may struggle with certain dialects or informal language usage, affecting its overall performance.

- **Contextual Challenges:** The model may face difficulties in understanding complex contexts or resolving ambiguities in Swahili text, especially in cases where entity references are subtle or context-dependent.

Significance of the Model:

The Swahili NER contributes significantly to the advancement of NLP tools for Swahili, addressing the need for better language processing capabilities in under-resourced languages. By improving NER for Swahili, the model supports greater inclusivity and accessibility in NLP technologies.

2.4. Summary

The MasakhaNER project and the Swahili NER model represent significant advancements in enhancing Named Entity Recognition (NER) for under-resourced languages. MasakhaNER's approach, using supervised learning, transfer learning, and multilingual models, demonstrates effectiveness in improving NER for ten African languages, though it faces challenges such as limited annotated data and inconsistencies in multilingual model performance. Similarly, the Swahili NER model leverages transfer learning with XLM-RoBERTa to fine-tune NER capabilities specifically for Swahili. Despite promising results, it also struggles with limited training data, language nuances, and contextual challenges. Both projects highlight the importance of creating inclusive NLP tools and set valuable precedents for similar efforts in other under-resourced languages.

WRITTEN BY |DJIOTSA DJOUAKE CHRISTIAN DARYN

III.   CHAPTER 3: ANALYSIS AND DESIGN

3.1.   Introduction

This chapter focuses on the design and analysis of the Named Entity Recognition (NER) system specifically developed for the Yemba language. NER is a crucial NLP task that involves identifying and classifying entities such as people, locations, dates, and quantity within text. For languages like Yemba, which are less represented in NLP research, developing an effective NER system poses unique challenges and opportunities.

We begin by detailing the methodology adopted for the Yemba NER system, including the collection and annotation of Yemba text data. This annotated data is fundamental for training the model and ensuring its accuracy. The chapter will outline the selection of appropriate machine learning techniques, including supervised learning and transfer learning, to enhance entity recognition capabilities for Yemba.

Next, we describe the global architecture of the Yemba NER system. This includes the design of the data processing pipeline, the integration of pre-trained models adapted for Yemba, and the overall system architecture that supports efficient entity recognition. The architecture is crafted to handle the specific linguistic features and challenges of Yemba, ensuring that the system performs well across various contexts.

We will also provide a detailed description of the algorithms used, including the training processes and fine-tuning techniques tailored for Yemba. Additionally, the chapter addresses the resolution process, focusing on methods for handling ambiguities and errors in entity recognition specific to the Yemba language.

By examining these elements, this chapter aims to offer a comprehensive understanding of how to design an effective NER system for under-resourced languages. The analysis highlights both the strengths and limitations of the chosen approaches, providing insights into the practical aspects of developing a high-performing NER system for a language with limited resources and research.

3.2. Methodology

The methodology for developing the Yemba Named Entity Recognition (NER) system leverages advanced deep learning techniques, focusing on the use of transformer architectures for complex learning tasks. By utilizing the pre-trained **"bert-base-multilingual-cased"** model, we can harness the power of a robust, multilingual transformer model to address the specific needs of Yemba NER.

Our approach centers on supervised fine-tuning of this pre-trained transformer model. The "bert-base-multilingual-cased" model, which has already been trained on a diverse set of languages, provides a strong foundation of linguistic knowledge. By fine-tuning this model with labeled Yemba data, we can adapt it to recognize and classify entities specific to the Yemba language.

This combination allows us to take full advantage of the pre-trained knowledge embedded in BERT, making the most of limited annotated data for Yemba. Fine-tuning the model ensures that it is tailored to handle the unique linguistic features and complexities of Yemba, thus enhancing its performance in identifying named entities accurately.

In summary, **our methodology integrates a deep learning approach using transformers with supervised fine-tuning of a pre-trained model**. This strategy enables us to develop an effective

NER system for Yemba, leveraging the strengths of BERT while addressing the challenges posed by limited labeled data.

3.3. Design

Below is an overview of the design process and key components that will be implemented to develop an effective NER system for the Yemba language.

3.3.1. Requirements

i. Requirements

System requirements:

- **Hardware:** High-performance computing resources such as GPUs for efficient training and fine-tuning of the transformer model.
- **Software:** Python, Tensorflow, Hugging Face Transformers library, Docker, and liberOffice Calc.

Data collection and preparation requirements:

- Collect a diverse and representative corpus of French text from various sources, including books, articles, and websites, for subsequent translation into Yemba by a professional Yemba Translator. The names of individuals are sourced from the student lists of the University of Dschang and the University of Yaoundé 1, while the locations are obtained from the 2005 data of the BUREAU CENTRAL DES RECENSEMENTS ET DES ÉTUDES DE POPULATION, as well as from various websites and articles.

- Ensure the corpus includes different contexts and uses cases to cover a wide range of entity occurrences.

ii. Data Annotation

- Manually annotate the collected corpus with entity labels such as **names of people**, **geopolitical location**(country, region, continents), **geographic location**(city, village, any location), **time**(date, days of the week, months of the year) and **quantity**(money, and percentage).
- Use libreOffice calc and Python code to facilitate the labeling process and ensure consistency and accuracy in the annotations**.**

iii. Preprocessing Pipeline

- **Text Normalization:** Clean and normalize the text to handle different forms of punctuation, capitalization, and other textual variations. Tokenize the text into words compatible with the BERT tokenizer.
- **Feature Extraction:** Convert the tokens into input features required by the BERT model, including token IDs, attention masks, and token type IDs.

iv. Model Selection and Fine-Tuning

- **Model Initialization:** Load the pre-trained **"bert-base-multilingual-cased"** model from the Hugging Face Transformers library. Initialize the model with a classification head for NER tasks.
- **Supervised Fine-Tuning:** Split the annotated dataset into training, validation, and test sets. Fine-tune the BERT model on the Yemba training data, adjusting hyperparameters to optimize performance. Hyperparameters consider here include **Learning Rate(**This

controls the step size the model takes when updating its weights during training**)** , **Batch Size(**this defines the number of data points processed before updating the model's weights**)** , **Number of Training Epochs(**This refers to the number of times the model iterates through the entire training dataset**).** And use validation data to monitor training progress and prevent overfitting.

v. Inference and Post-Processing

- **Entity Recognition:** Apply the fine-tuned model to new Yemba texts to identify and classify named entities. And output the recognized entities along with their labels.
- **Post-Processing:** Implement rules or heuristics to refine and validate the recognized entities. And Handle cases of ambiguity or errors by incorporating feedback mechanisms to improve the model over time.

vi. Evaluation and Feedback Loop

- **Model Evaluation:** Evaluate the model's performance using metrics such as precision, recall, and F1 score on the test dataset. And Analyze the results to identify areas for improvement.
- **Continuous Improvement:** Incorporate feedback from users and linguistic experts to refine the model. Regularly update the training data and fine-tune the model to adapt to new linguistic patterns and usage.

3.4.   Global Architecture of the Yemba Named Entity Recognition

The global architecture of the Yemba Named Entity Recognition (NER) system is designed to effectively handle the complexities of recognizing and classifying named entities in Yemba text.

This architecture integrates several key components, each playing a vital role in ensuring the system's accuracy, efficiency, and usability.

3.4.1. Data ingestion

At the heart of the architecture is a robust data ingestion mechanism that facilitates the collection and preprocessing of Yemba text data. This ensures that the data is in the appropriate format for further processing. The preprocessing pipeline then prepares this data through normalization, tokenization, and feature extraction, setting the stage for effective model training.

3.4.2. NER Model

The core of our system is a fine-tuned transformer-based NER model, specifically utilizing the "**bert-base-multilingual-cased**" architecture. This model is adapted to the Yemba language through supervised fine-tuning, leveraging pre-trained knowledge to improve its performance on Yemba-specific tasks.

3.4.3. Post-Processing

Following entity recognition, a comprehensive post-processing phase is implemented to refine and validate the model's output. This involves techniques to handle ambiguities and errors, ensuring the final results are accurate and reliable.

3.4.4. User interface

Additionally, an optional user interface is included to enhance the system's functionality. This interface provides tools for data entry, manual annotation, and feedback collection, allowing users to interact with the system and contribute to its continuous improvement by providing real life Yemba text.

3.5. Description of the Resolution Process for the Yemba Named Entity Recognition System

The resolution process in the Yemba Named Entity Recognition (NER) system is crucial for refining the model's outputs, handling inaccuracies, and ensuring the system's overall effectiveness. It comprises a series of systematic steps designed to enhance entity recognition, resolve ambiguities, and integrate user feedback. Below is a detailed description of the resolution process:

- **Entity Validation:** Verify that the entities recognized by the NER model are correct and consistent with the expected labels. Using techniques as rule-based checks to confirm that recognized entities adhere to expected patterns.

- **Ambiguity Resolution:** Resolve cases where the context is insufficient to clearly identify the entity type.

- **Error Handling and Correction:** Identify, address, and correct errors in entity recognition to improve model accuracy.

- **Feedback Integration:** Use feedback from users and experts to continuously improve the model's performance.

The resolution process for the Yemba NER system is a comprehensive approach designed to enhance the quality of entity recognition. By validating entities, resolving ambiguities, handling errors, integrating feedback, and monitoring performance, the process ensures that the system delivers accurate and reliable results. This iterative and adaptive approach is essential for

continuously improving the Yemba NER system and maintaining its effectiveness in real-world applications.

3.6. Summary

The methodology outlined for the Yemba Named Entity Recognition (NER) system underscores a strategic approach to developing an effective tool for identifying and classifying entities in the Yemba language. By leveraging advanced deep learning techniques, specifically the transformer architecture with the "bert-base-multilingual-cased" model, the system benefits from a robust foundation of pre-trained linguistic knowledge. This foundation is further tailored to the Yemba language through supervised fine-tuning, allowing the model to handle the unique features and complexities of Yemba with improved accuracy despite the constraints of limited annotated data.

The design and implementation of the Yemba NER system involve several key components: a comprehensive **data ingestion** and **preprocessing pipeline**, a **fine-tuned transformer-based NER model**, and a **post-processing phase** for refining the results. Other features, such as a **user interface**, provide additional functionality for annotation, visualization, and feedback collection, enhancing the system's usability and effectiveness.

The resolution process, a critical aspect of the system, **ensures the accuracy and reliability of entity recognition through validation, ambiguity resolution, error correction, and continuous feedback integration**. This iterative and adaptive approach is vital for addressing challenges and maintaining the system's performance over time.

In summary, the combination of sophisticated modeling techniques and a well-structured design and resolution process positions the Yemba NER system as a valuable tool for advancing natural language processing capabilities in the Yemba language, paving the way for more inclusive and effective linguistic technologies.

CHAPTER 4: DATASET CONSTRUCTION

4.1. Introduction

The construction of a dataset is crucial for the development of an effective Named Entity Recognition (NER) system for the Yemba language. In this chapter, we outline the comprehensive process involved in constructing the dataset used for training and evaluating the Yemba NER system. The dataset was developed by translating French sentences into Yemba, ensuring that the text accurately represents the linguistic and contextual features necessary for effective entity recognition.

4.2. Requirements

To create a robust dataset, several requirements must be met:

- **Hardware:** Sufficient computational resources to handle data processing and annotation tasks. And to ensure that we make use of Kaggle technology which is one of the world's largest data science communities with powerful tools and resources to help data scientists achieve their science goals.
- **Software:** Python for dataset augmentation algorithms, LibreOffice Calc to visualize the dataset and for validation of the dataset.
- **Human Resources:** Skilled translators for accurate French-to-Yemba translation and annotators for labeling entities.

4.3. Definition of Entities Types

Defining the types of entities is essential for accurate annotation and effective NER model training. For the Yemba dataset, we classify entities into the following types:

- **People:** his entity type encompasses names of individuals found within the dataset. To ensure a comprehensive representation, names were sourced from a variety of reliable and relevant channels: **official university records and published materials**. Names were extracted from student lists at the University of Dschang and the University of Yaounde 1. This approach guarantees the inclusion of authentic and current Yemba names commonly used in the region. Additionally, names were obtained from reputable books and websites written in Yemba. This broadens the name pool and captures variations in Yemba naming conventions used in different contexts.

- **Locations:** Geopolitical locations (countries, regions, continents) and geographic locations (cities, villages, any specific places). Location entities were extracted from "BUREAU CENTRAL DES RECENSEMENTS ET DES ÉTUDES DE POPULATION". And the remaining extracted from websites and articles.

- **Time:** Temporal entities such as dates, days of the week, and months of the year.

- **Quantity:** Entities related to measurements, such as money as percentages, and distance.

4.4. Data Selection and Collection

4.4.1. Data Selection

Data selection involves identifying and choosing appropriate French sentences for translation into Yemba. This includes:

- **Source Variety:** Selecting sentences from diverse contexts and domains, such as literature, news articles, and everyday conversations, to ensure a representative corpus.
- **Relevance:** Ensuring that selected sentences contain a wide range of entities and linguistic structures relevant to the Yemba language.

4.4.2. Data Collection

The collection process includes:

- **Translation:** Translating selected French sentences into Yemba. This is achieved through manual translation by skilled translators to maintain accuracy and context.
- **Corpus Assembly:** Compiling the translated sentences into a comprehensive corpus that represents various entity types and contexts.

4.5. Data Preparation

Data preparation involves processing the collected data to make it suitable for annotation and model training by performing a **Text Normalization(**Standardizing the text to handle variations in punctuation, capitalization, and other textual features.)

4.6. Data Annotation

Annotation is a critical step where entities in the Yemba text are labeled according to predefined types:

- **Manual Annotation:** Annotators manually label the entities in the Yemba text using tools such as LibreOffice Calc and custom Python scripts to ensure accuracy to speed up the process by automating some entities annotation.
- **Annotation Guidelines:** Providing clear guidelines for annotators to maintain consistency in entity labeling.

4.7. Data Augmentation

To enhance the dataset and improve model robustness, we apply data augmentation techniques:

4.7.1. Entities Replacements

Entity replacements are a valuable technique to enhance the variability and coverage of your Yemba named entity recognition (NER) dataset. This approach involves strategically replacing specific entities with alternative entities that share the same entity type.

This technique involves identifying existing entities within your dataset and replacing them with alternative entities of the same type. For example, if the dataset contains the named location "Dschang," you could replace it with another university name like "Ngaoundéré.".

Benefits of Entity Replacements:

- **Increased Variability:** By introducing alternative entities, you expose the NER model to a broader range of variations within each entity type. This improves its ability to generalize and recognize similar entities in unseen data.
- **Enhanced Coverage:** Replacing entities helps expand the dataset's coverage of potential entities the model might encounter. This leads to a more robust NER model capable of handling a wider variety of real-world Yemba text.

Considerations:

- **Maintaining Accuracy:** It's crucial to ensure the replacement entities remain accurate representations of the original entity type.

- **Domain Specificity:** Consider the specific domain of your NER task. If focused on news articles, replacing locations with fictional names might not be ideal. Choose replacements that are relevant to the expected domain of the data.

4.7.2.   Sentence Concatenation with Logic Connectors

Sentence concatenation with logic connectors is a creative approach to address the challenge of limited data in Named Entity Recognition (NER) for under-resourced languages. NER datasets often lack complex sentence structures and diverse entity relationships found in real-world text. This can limit the model's ability to generalize to unseen data.

Sentence concatenation involves merging existing sentences within your dataset using logic connectors (AND, OR, NOT) to create new, more complex sentences. This approach aims to **increase dataset size(**Artificially expand the dataset by creating new variations of existing data points.**)** and **introduce complex structures(**Simulate real-world text patterns with longer sentences and potentially richer entity relationships.**)**

For a good quality dataset we decide to combine both techniques to ensure entity variety and to increase the number of entities that can be injected to the dataset. To do that we used ten Yemba sentence connectors and the 231 sentences that the translators were able to translate.

Therefore, the data augmentation algorithm was able to generate from n sentences, k sentences connectors **n(n-1)*k augmented labels.** which give us **531300 augmented labels.** Also, **both sentence concatenation and entity replacement were combined**. Ensuring the variety of entities within the dataset(such even entities of the same type but not of the same number of words can be replaced effectively).

Below is the list of Yemba sentence connectors used:

YEMBA	Pá'	tétá'	Tɛ́mbɔ	ké	pɔ́	ŋ́ghʉɛ̄	Méla'mi	Ánɛ	nnɔŋɔ	Elɛkɔ̄
ENGLISH	as/ like	but	then	or	and	and	because	on	in	comment

4.8. Dataset Splitting

Splitting the dataset into distinct subsets for training, validation, and testing is crucial for evaluating model performance:

- **Training Set:** Used for training the model and learning entity recognition patterns.
- **Validation Set:** Employed to tune model parameters and prevent overfitting.
- **Test Set:** Assesses the model's generalization and performance on unseen data.

4.9. Summary

The construction of the Yemba NER dataset involved translating French sentences into Yemba, carefully building sentences to reflect diverse contexts, meticulous data preparation and manual annotation for accuracy, and strategic data augmentation techniques such as entity replacements and logical connectors to enhance variability, ultimately ensuring a comprehensive and high-quality resource for training and evaluating named entity recognition models.

CHAPTER 5: IMPLEMENTATION AND RESULTS

5.1. Introduction

Implementing the Yemba Named Entity Recognition (NER) system involves several critical steps to ensure the model's effectiveness in accurately identifying and classifying named entities within Yemba text. This chapter outlines the implementation process, detailing the integration of advanced machine learning techniques, the utilization of pre-trained models, and the application of fine-tuning to adapt the model to the Yemba language.

5.2. Tools and Materials Used

For the successful implementation of the Yemba Named Entity Recognition (NER) system, a variety of tools and platforms are utilized to ensure efficiency, accessibility, and maintainability. The following are the key tools and materials employed in this project:

- **Kaggle:** Provides a comprehensive notebook environment for code editing, 12 hours of running time per session, and 30 hours of GPU usage per week, significantly accelerating the NER training process. Additionally, it offers a platform to host our dataset, facilitating easy access during model training and data augmentation.

- **TensorFlow:** Used for building and fine-tuning the NER model, offering extensive support for deep learning and machine learning tasks.

- **Hugging Face Hub:** Hosts the fine-tuned NER model and deploys it as a FastAPI, making the model accessible globally for various applications.

- **GitHub:** Tracks all code versions, ensuring seamless deployment and maintainability of the codebase.

- **Transformers Library:** Provides pre-trained models and tools necessary for leveraging transformer architectures in NER tasks.

- **BERT Pretrained Model**: Specifically, the **"bert-base-multilingual-cased"** model is utilized for its robust multilingual capabilities, serving as the foundation for fine-tuning on Yemba data.

- **Python:** The primary programming language used for developing the NER system, due to its extensive libraries and community support for machine learning and data processing.

- **Django REST Framework:** Used for developing the backend of the Yemba NER web-based application, ensuring a robust and scalable API.

- **Docker:** Utilized for containerizing the application, enabling consistent and reliable deployment across different environments.

- **HTML, CSS, and JavaScript:** Employed for developing the front end of the web-based application, providing an interactive user interface for data entry, annotation, and feedback.

- **EmailJS:** Integrated for custom feedback collection, allowing users to provide insights and suggestions directly through the application.

- **PostgreSQL:** Chosen as the database solution for storing annotated data, user inputs, and other relevant information, ensuring data integrity and efficient retrieval.

- **Render:** A deployment platform for hosting both the backend and frontend of the Yemba NER application, ensuring smooth and reliable access to the system.

By leveraging these tools and materials, the Yemba NER project ensures a comprehensive, scalable, and user-friendly system capable of accurately identifying and classifying named entities in Yemba text.

5.3. Description of the Implementation Process

The implementation of the Yemba Named Entity Recognition (NER) system involves a structured and methodical approach, ensuring each component is developed, integrated, and optimized for performance. Below is a detailed description of the implementation process:

5.3.1. Environment Setup

- **Kaggle:** Set up a notebook environment on Kaggle, leveraging its computational resources for model training. This includes configuring the GPU settings and ensuring necessary libraries are installed.

```
!pip install huggingface_hub evaluate seqeval openpyxl ipywidgets, tensorflow
!pip install --upgrade wandb
!pip install --upgrade tensorflow
!pip install --upgrade grpcio
```

Figure 1: Install dependencies

```
: from huggingface_hub import login, notebook_login
  # Log in to Hugging Face
  notebook_login()
```

Figure 2: login to Huggin Face Hub

```
import pandas as pd
import numpy as np
import ast
import evaluate
import tensorflow as tf
from sklearn.model_selection import train_test_split
from transformers import TFAutoModelForTokenClassification, AutoTokenizer, create_optimizer, DataCollatorForTokenCla
from datasets import Dataset, load_dataset
import wandb
```

Figure 3: Import required libraries

5.3.2. Data Handling

- **Data Collection and Preparation**

- **Data Annotation**

```
# Function to load data
def load_data(input_file):
    all_sheets = pd.read_excel(input_file, sheet_name=None, engine='openpyxl')
    combined_df = pd.concat(all_sheets.values(), ignore_index=True)
    return combined_df
```

Figure 4: Load dataset function

3. Preprocess data

```python
def preprocess_data(df):
    """Converts tag column to uppercase using vectorized operations.
    Args:
      df (pd.DataFrame): Input DataFrame.
    Returns:
      pd.DataFrame: DataFrame with uppercase tags.
    """
    def convert_tags_to_uppercase(tags):
        if isinstance(tags, str):
            tags_list = ast.literal_eval(tags)
        else:
            tags_list = tags
        return [tag.upper().replace('PÓB-PER', 'B-PER').replace('I-QUANT', 'I-QUAN').replace('B-TIMELĀ', 'B-TIME').

    df['Tag'] = df['Tag'].apply(convert_tags_to_uppercase)
    return df
```

Figure 5: Preprocess loaded dataset

```python
# Token and label creation functions
def create_tokens(text):
    return [word for word in text.split()]

def create_num_labels(label):
    return [label2index[text] for text in label]

# Function to tokenize and align labels
def tokenize_and_align_labels(examples):
    tokenized_inputs = tokenizer(examples["Tokens"], truncation=True, is_split_into_words=True)
    labels = []
    for i, label in enumerate(examples[f"NER_Tags"]):
        word_ids = tokenized_inputs.word_ids(batch_index=i)
        previous_word_idx = None
        label_ids = []
        for word_idx in word_ids:
            if word_idx is None:
                label_ids.append(-100)
            elif word_idx != previous_word_idx:
                label_ids.append(label[word_idx])
            else:
                label_ids.append(-100)
            previous_word_idx = word_idx
        labels.append(label_ids)
    tokenized_inputs["labels"] = labels
    return tokenized_inputs
```

Figure 6: Tokenization

WRITTEN BY |DJIOTSA DJOUAKE CHRISTIAN DARYN

6. Load and preprocess data

```
|: import pandas as pd
   # Load and preprocess data
   input_file = '/kaggle/input/final-yemba-dataset/yembaner_dataset.xlsx' # Replace with your actual file path
   data = load_data(input_file)
   data.head()
```

```
|: import ast
   # preprocess data
   data = preprocess_data(data)

   print("Data preprocessed!")
```

Figure 7: Loading and Preprocessing the data

7. Define labels and their mappings ¶

```
# Define labels and their mappings
label2index = {"O":0,"B-PER":1,"I-PER":2,"B-GEO":3,"I-GEO":4,"B-QUAN":5,"I-QUAN":6, "B-GPE":7,"I-GPE":8,"B-TIME":9,
index2label = {v:k for k,v in label2index.items()}
label_names = [key for key in label2index.keys()]
```

Figure 8: Define and mapping Lables

8. Create tokens and numeric labels

```
|: # Create tokens and numeric labels
   data['Tokens'] = data['Sentences'].apply(lambda x: create_tokens(x))
   print('Sentence tokens added!')
```

```
|: data['NER_Tags'] = data['Tag'].apply(lambda label: create_num_labels(label))
   print('Tags numiric value added added!')
```

9. Remove rows with unequal # tokens and # tags

```
|: # Remove rows with unequal # tokens and # tags
   index_labels = []
   for i in range(len(data)):
       if len(data['Tokens'][i]) != len(data['NER_Tags'][i]):
           print(f"Tokens and tags at index {i} don't match")
           index_labels.append(i)
   data.drop(index=index_labels, inplace=True)
   data.reset_index(drop=True, inplace=True)
```

```
|: print("dataset size", len(data))
```

Figure 9: Prepare the data for the training.

5.3.3. Model Development

- **Model Initialization**: Load the pre-trained "bert-base-multilingual-cased" model from the Hugging Face Transformers library.

- **Splitting the train and test dataset into Chunk for more resourced hardware resource management and training the model incrementally.**

- **Supervised Fine-Tuning:** Fine-tune the pre-trained BERT model using the annotated Yemba dataset. This involves splitting the dataset into training, validation, and test sets. Optimize hyperparameters such as learning rate, batch size, and the number of training epochs to enhance model performance.

- **Training Process:** Utilize Kaggle's GPU resources to accelerate the training process. Monitor training progress and validation performance to prevent overfitting and ensure the model generalizes well to unseen data.

10. Split the dataset into chunks of max 100000 lines each

```
]:  # Split the dataset into chunks of 10000 lines each
    chunk_size = 100000
    num_chunks = len(data) // chunk_size + (1 if len(data) % chunk_size != 0 else 0)

    model_name = 'bert-base-multilingual-cased'
    tokenizer = AutoTokenizer.from_pretrained(model_name)
    data_collator = DataCollatorForTokenClassification(tokenizer=tokenizer)
    seqeval = evaluate.load("seqeval")
```

Figure 10: Split the dataset into chunks and the tokenizer

11. Initialize the model

```
]:  from transformers import AutoModelForTokenClassification
    model = AutoModelForTokenClassification.from_pretrained(
        model_name, num_labels=len(label_names), id2label=index2label, label2id=label2index
    )
```

Figure 11: Initialize the pretrained model

12. Training arguments (model hyperparametter)

```python
[ ]: from transformers import TrainingArguments
     # Training arguments
     training_args = TrainingArguments(
         output_dir="/kaggle/working/ner_model",
         learning_rate=2e-5,
         per_device_train_batch_size=9,
         per_device_eval_batch_size=6,
         num_train_epochs=3,
         weight_decay=0.01,
         eval_strategy="epoch",
         save_strategy="epoch",
         load_best_model_at_end=True
     )
```

Figure 12: Set train *Hyper Parametters*

```python
from transformers import Trainer

# Iterate through each chunk and train the model incrementally
for i in range(num_chunks):
    print(f"Processing chunk {i+1}/{num_chunks}")
    chunk_data = data.iloc[i*chunk_size:(i+1)*chunk_size]
    chunk_dict = chunk_data[['Tokens', 'NER_Tags']].to_dict('list')
    raw_dataset = Dataset.from_dict(chunk_dict)

    split = raw_dataset.train_test_split(test_size=0.1, shuffle=True, seed=42)
    tokenized_dataset = split.map(tokenize_and_align_labels, batched=True)

    #    normal trainer
    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=tokenized_dataset["train"],
        eval_dataset=tokenized_dataset["test"],
        tokenizer=tokenizer,
        data_collator=data_collator,
        compute_metrics=compute_metrics,
    )

    trainer.train()


    model_dir = f"/kaggle/working/ner_model/yembaner_chunk_{i+1}"
    trainer.save_model(model_dir)
    tokenizer.save_pretrained(model_dir)

    # Push the model to the Hugging Face Hub
    model.push_to_hub("MHULO/yembanerlive")
    tokenizer.push_to_hub("MHULO/yembanerlive")

    # Load the best model for the next chunk
    model = AutoModelForTokenClassification.from_pretrained(model_dir)
```

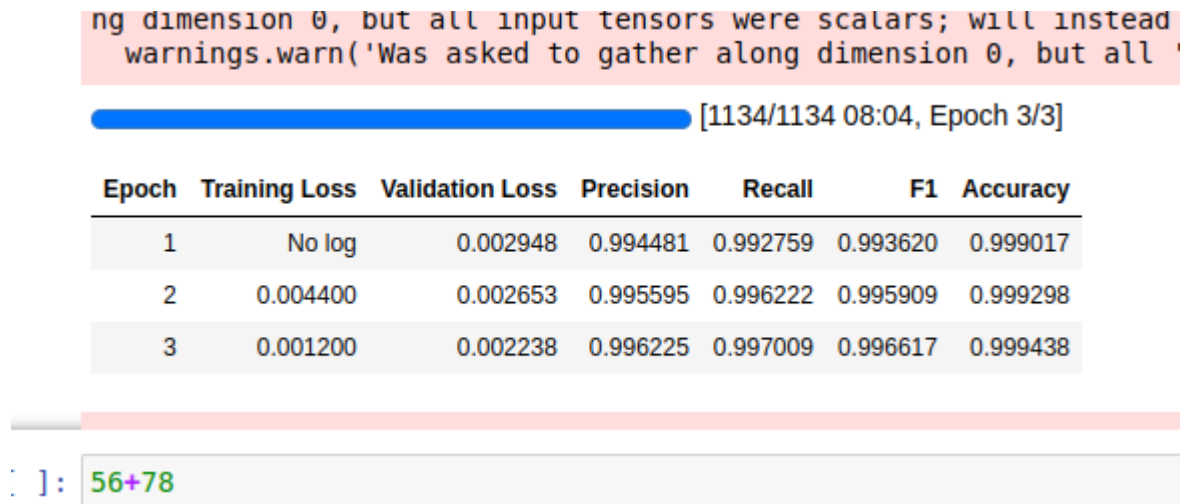*Figure 13: Iterate Through each chunk and train the yemba NER*

encrementally

```
ng dimension 0, but all input tensors were scalars; will instead
    warnings.warn('Was asked to gather along dimension 0, but all '
```

[1134/1134 08:04, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1	No log	0.002948	0.994481	0.992759	0.993620	0.999017
2	0.004400	0.002653	0.995595	0.996222	0.995909	0.999298
3	0.001200	0.002238	0.996225	0.997009	0.996617	0.999438

```
]: 56+78
```

Figure 14: Training metrics

5.3.4. Inference and Post-Processing

- **Entity Recognition:** Apply the fine-tuned model to new Yemba texts to identify and classify named entities. Output the recognized entities along with their respective labels.

- **Post-Processing:** Implement rules and heuristics to refine and validate the recognized entities. Address ambiguities and errors by incorporating feedback mechanisms to improve model accuracy over time.

5.3.5. System Integration

- **Backend Development:** Develop the backend of the Yemba NER web application using Django REST Framework. Create APIs for model inference, data management, and user feedback.

- **Frontend Development:** Build an interactive user interface using HTML, CSS, and JavaScript. The interface includes tools for data entry, manual annotation, visualization of recognized entities, and feedback collection.

- **Containerization:** Use Docker to containerize the model, ensuring consistent deployment across different environments.

5.3.6. Deployment and Maintenance

- **Model Hosting:** Host the fine-tuned model on Hugging Face Hub, deploying it using the docker container as a FastAPI endpoint for global accessibility.

- **Application Deployment:** Deploy the backend and frontend services on Render, ensuring the application is accessible to users.

- **Continuous Improvement:** Regularly update the training data and fine-tune the model based on user feedback. Use GitHub to track code changes and manage version control, facilitating ongoing maintenance and enhancements.

5.4. Presentation and Interpretation of YembaNER

5.4.1. Home page

This page serves as a landing page and provides information that will help the user to use the service efficiently.
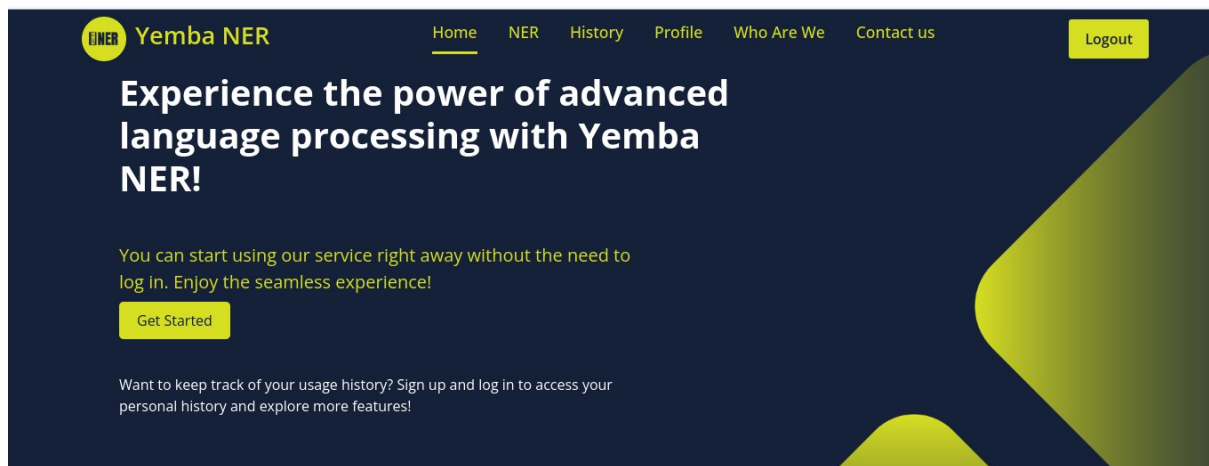
Figure 15: Home page

5.4.2.   NER page

This page serves as the entry point for users to make predictions using the Yemba NER model by providing a text and then get their entities named. Also If the user is logged in, the output of his prediction along with the input text will be saved to the database and he could get it from the History page.
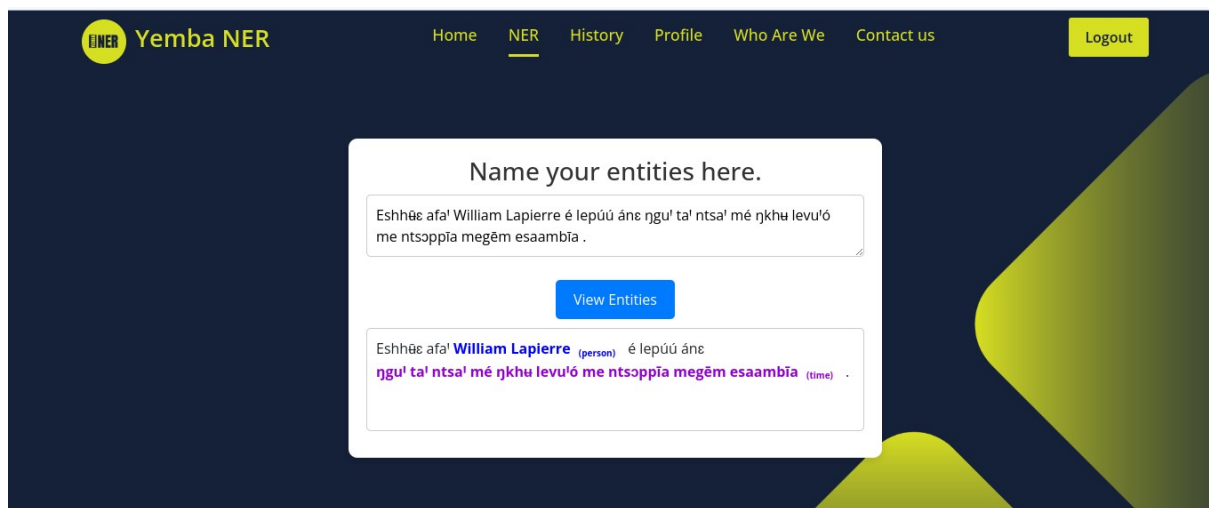
Figure 16: NER page

5.4.3. History page

The history page returns the previous predictions performed by the user such that the recents ones are the top of the table.
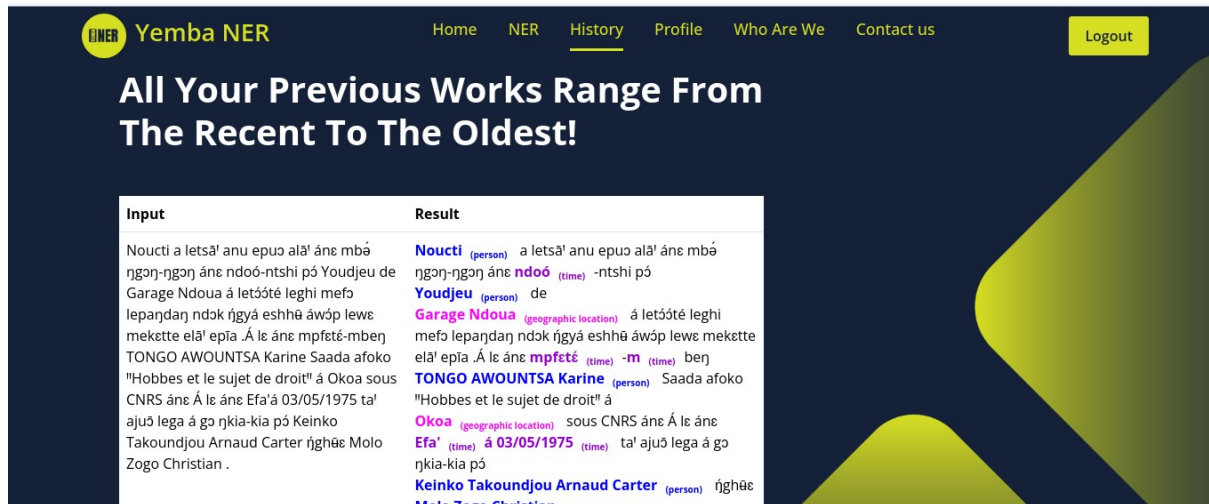
Figure 17: History page

5.4.4. Login panel

This panel allows users to connect to their existing account such that they can benefit from the permission to save their prediction and see their profile. The will need to provide a valid email address and password.
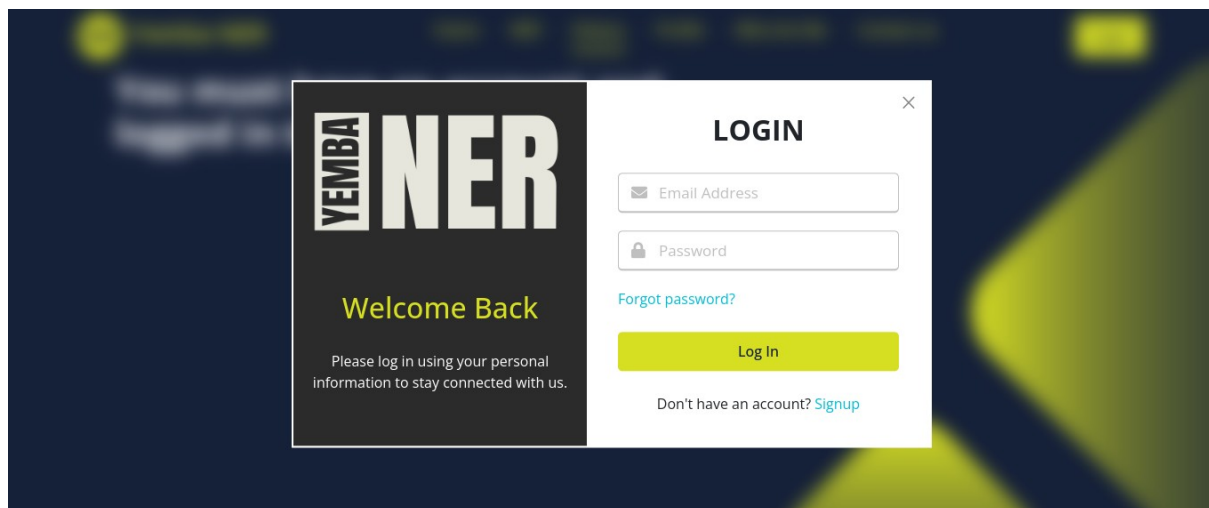
Figure 18: Login panel

5.4.5. Signup panel

This panel serves as an entry point for users to register to the Yembar NER platform. The users should provide their email address (which is used for email validation), username, first name, last name and password for authentication.
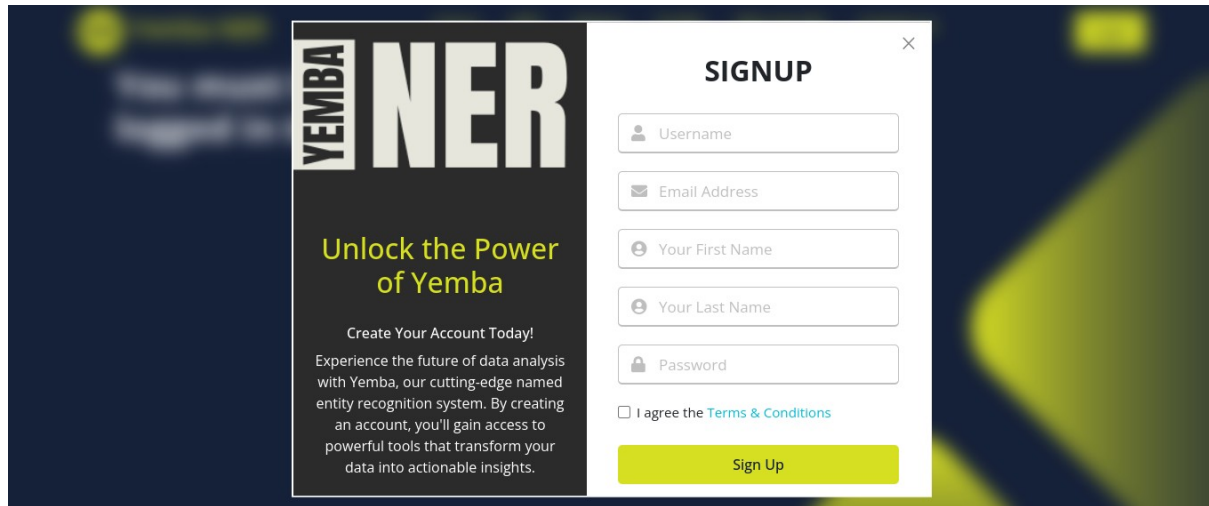
Figure 19: Signup panel

5.4.6. User profile

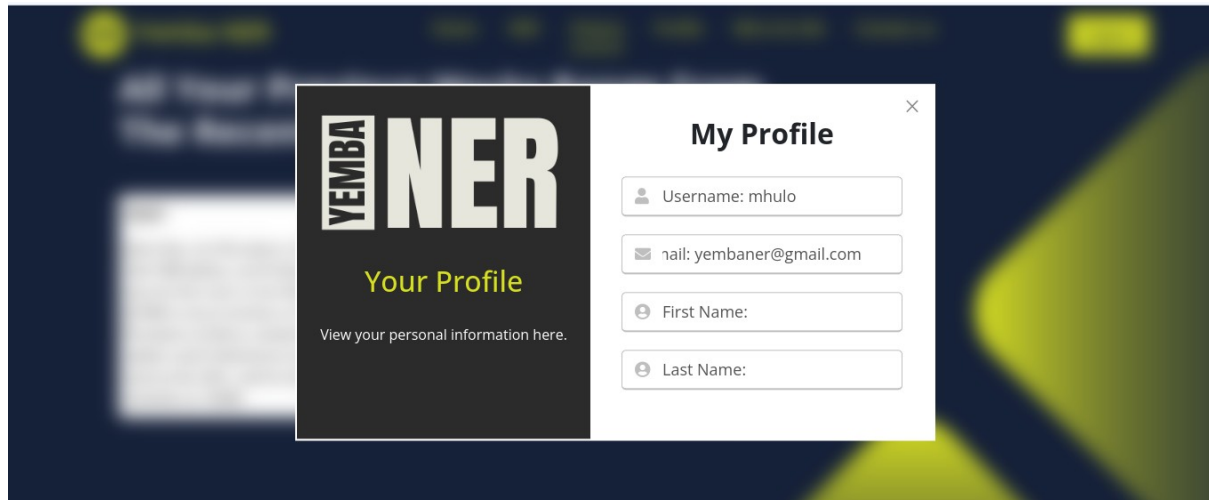Provides the user with his credentials if the user is logged in.

Figure 20: User Profile Panel

5.4.7. Reset password

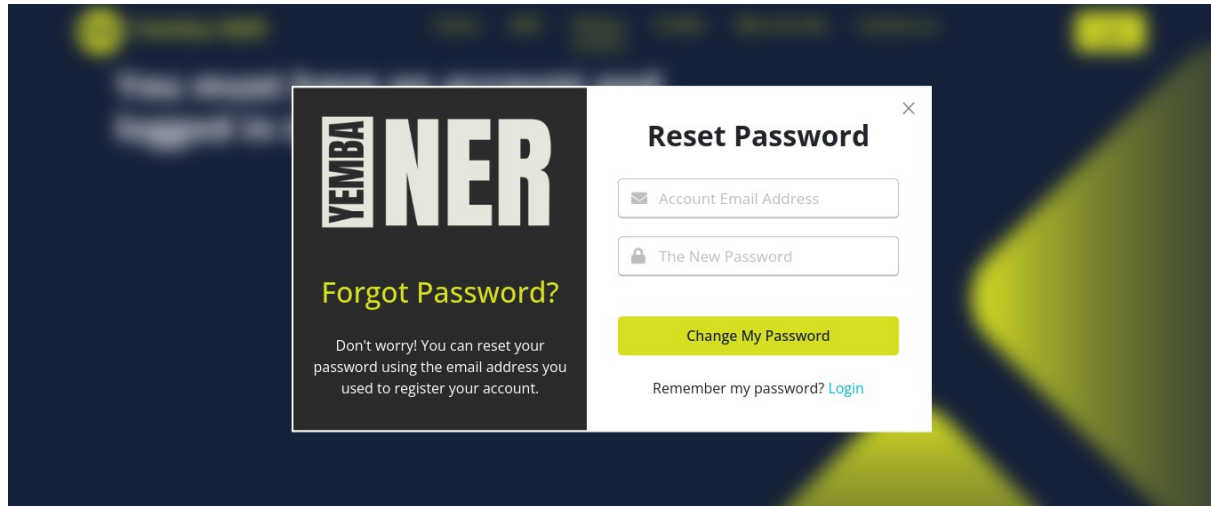Also, the platform allows users to reset their password using their email if forgotten.

Figure 21: Password reset panel

5.4.8. About us

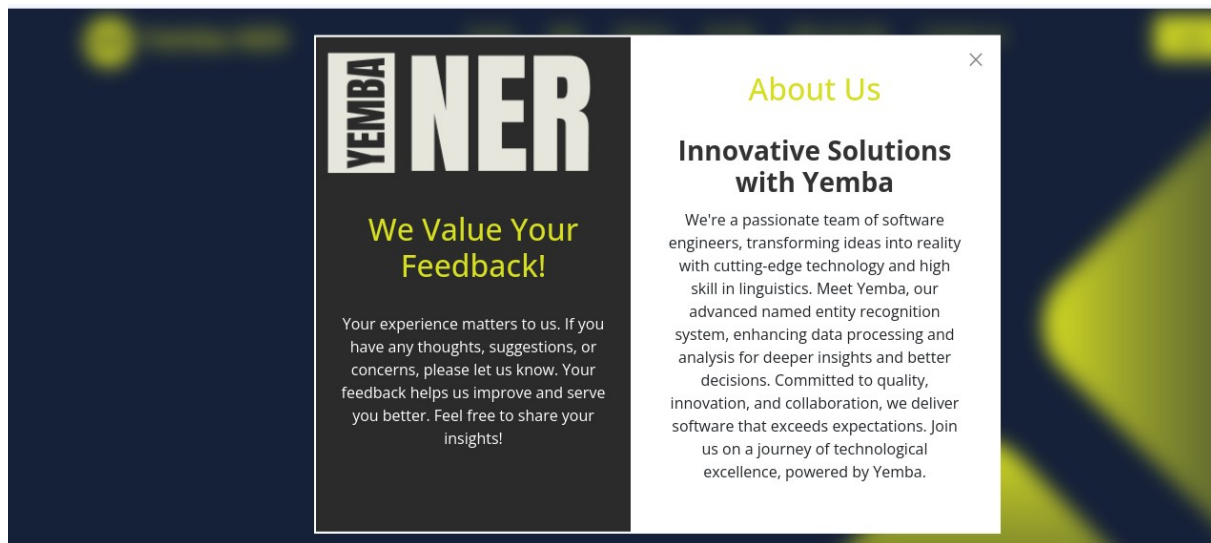This panel provides users with information about the Yemba NER platform.

Figure 22: About us

WRITTEN BY |DJIOTSA DJOUAKE CHRISTIAN DARYN

5.4.9. Contact us and feedback

This panel plays a very important role providing user support and allowing users to send feedback about their point of view of the application which enables a continuous update of the Yemba NER and a better product for the future.
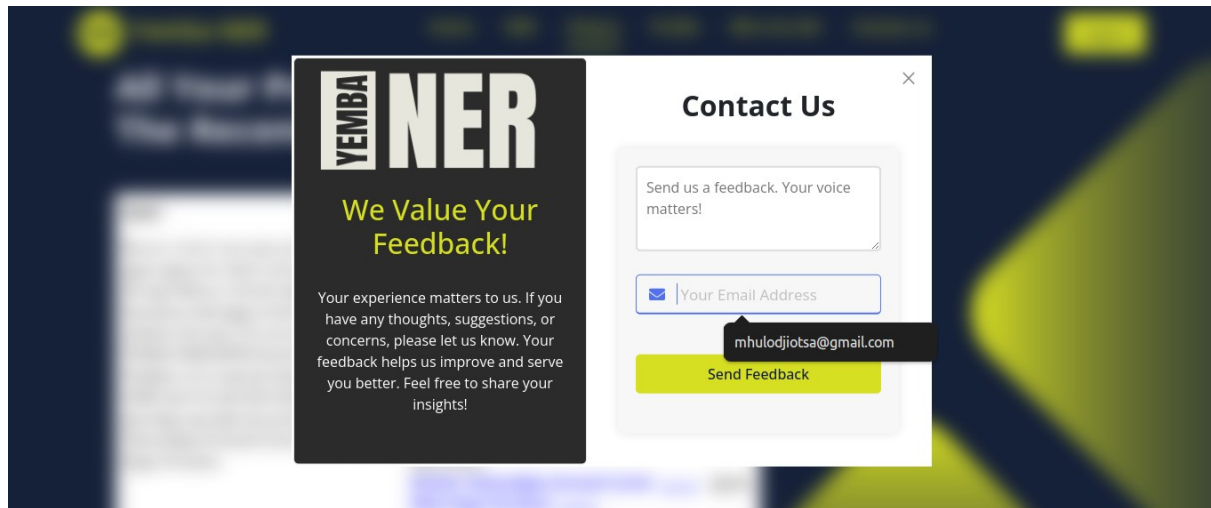

Figure 23: Contact Us

5.5. Evaluation of the YembaNER

The evaluation is positive! The model has been successfully trained and demonstrates the ability to identify four key entity types: person, locations, time and quantity. This indicates a strong foundation for real-world NER tasks.

However, some room for improvement exists:
- **Incomplete Entity Recognition:** The model occasionally struggles to recognize certain entities within the data. Further investigation is needed to pinpoint the specific entity types or data patterns causing these issues.

- **False Positives:** In some cases, the model might incorrectly identify non-entity words or phrases as entities. This can lead to inaccurate results and requires refinement.

5.6. Summary

Overall, the implemented Yemba NER model shows promising results for Yemba named entity recognition. By addressing the identified limitations and implementing the suggested improvements, the model's accuracy and reliability can be further enhanced.

CHAPTER 6: CONCLUSION AND FURTHER WORKS

6.1. Summary of Findings

The Yemba Named Entity Recognition (NER) project successfully established a foundational model for identifying key entities (people, locations, times, quantities) within Yemba text. This accomplishment involved leveraging computational resources and assembling a team of translators, annotators, and language specialists. By combining human expertise with machine learning techniques, a Yemba dataset was constructed and subsequently used to train a deep learning model. While the model demonstrates promising results, further refinements are planned to address specific entity recognition challenges and minimize false positives. This project lays the groundwork for future advancements in Yemba language processing.

6.2. Contribution to Engineering and Technology

Overall, the Yemba NER project contributes to engineering and technology by promoting Yemba language accessibility, advancing NLP techniques for under-resourced languages, and laying the foundation for future Yemba NLP projects. It's a stepping stone towards a more inclusive technological landscape that caters to the needs of diverse language communities.

6.3. Recommendations

The Yemba Named Entity Recognition (NER) project demonstrates the significant impact that Natural Language Processing (NLP) can have on under-resourced languages. By successfully building a foundational model for identifying key entities within Yemba text, this project opens doors for broader technological integration of the language.

Therefore, it is strongly recommended that the Faculty of Engineering and Technology prioritize investment in similar projects focused on under-resourced languages. Here's why:

- **Empowering Under-Resourced Languages:** NLP advancements like Yemba NER empower these languages by enabling the development of crucial applications. These can include machine translation tools, information retrieval systems, or Yemba-specific chatbots. Such applications bridge the digital divide for speakers, fostering greater access to technology and information.

- **Advancing the Field of NLP:** Under-resourced language projects like Yemba NER present valuable challenges that can propel the field of NLP forward. The project's development process, from addressing limited data resources to tackling specific entity recognition issues, informs the creation of more robust NLP techniques. This knowledge benefits the broader NLP community and contributes to more inclusive language technologies.

- **Fostering Innovation and Collaboration:** Investing in under-resourced language NLP fosters innovation and collaboration between engineers, linguists, and language specialists. These collaborations can lead to groundbreaking advancements not just for specific languages, but for the entire field of NLP and its applications.

The Yemba NER project serves as a successful example, paving the way for further exploration in this domain. By investing in similar projects, the Faculty of Engineering and Technology can position itself at the forefront of inclusive technological development, empowering under-resourced languages and shaping the future of NLP.

6.4. Difficulties Encountered

While the Yemba Named Entity Recognition (NER) project achieved significant progress, it also encountered some challenges inherent to working with under-resourced languages:

51

- **Inexistent of Data Resources:** A significant hurdle was the limited availability of Yemba text data for training the NER model. This scarcity of data can hinder the model's ability to learn and recognize diverse entity variations effectively.

- **Data Annotation Challenges:** Annotating Yemba text for entity recognition requires expertise in both the Yemba language and NER principles. Finding qualified annotators and ensuring consistent annotation quality can be a challenge, especially for a less common language.

- **Specific Entity Recognition Issues:** The model might struggle with recognizing certain entity types within Yemba text. This could be due to factors like the lack of specific Yemba named entity conventions or limited training data for those particular entities.

- **Balancing Accuracy and Generalizability:** Striking a balance between model accuracy on the training data and its ability to generalize effectively to unseen Yemba text can be difficult. Overfitting to the training data can occur if the dataset is limited, resulting in poor performance on unseen data.

6.5. Further works

Building upon the successes of the Yemba NER project, several avenues exist for further exploration:

- **Data Augmentation Techniques:** Implementing techniques like back-translation (translating Yemba text to another language and back) or synonym replacement can artificially expand the dataset and improve the model's robustness in handling unseen data variations.

- **Active Learning:** Exploring active learning approaches where the model identifies the most informative data points for further annotation can be beneficial, especially when dealing with limited initial datasets.

- **Ensemble Learning:** Investigating the use of ensemble learning, which combines multiple NER models with different strengths, can potentially lead to improved overall performance and robustness.

- **Domain-Specific NER Models:** Developing NER models tailored to specific domains like news articles or medical documents can further enhance accuracy and cater to the needs of specialized applications.

REFERENCES

Endangered languages project—Search. (n.d.). Retrieved May 5, 2024, from https://www.endangeredlanguages.com/lang/country/Cameroon

Kandybowicz Jason and Harold Torrence. (n.d.). *Africa's endangered languages*. https://global.oup.com/academic/product/africas-endangered-languages-9780190256340?cc=us&lang=en& (Original work published 2017)

Yemba in cameroon. (n.d.). UNESCO WAL. Retrieved May 5, 2024, from https://en.wal.unesco.org/countries/cameroon/languages/yemba

Adelani, D. I., Abbott, J., Neubig, G., D'souza, D., Kreutzer, J., Lignos, C., Palen-Michel, C., Buzaaba, H., Rijhwani, S., Ruder, S., Mayhew, S., Azime, I. A., Muhammad, S., Emezue, C. C., Nakatumba-Nabende, J., Ogayo, P., Aremu, A., Gitau, C., Mbaye, D., … Osei, S. (2021). *Masakhaner: Named entity recognition for african languages* (arXiv:2103.11811). arXiv. https://doi.org/10.48550/arXiv.2103.11811

Horev, R. (2018, November 17). *BERT Explained: State of the art language model for NLP*. Medium. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

How docker containers work – explained for beginners. (2023, October 23). freeCodeCamp.Org. https://www.freecodecamp.org/news/how-docker-containers-work/

Mbeukman/xlm-roberta-base-finetuned-ner-swahili · hugging face. (2024, January 18). https://huggingface.co/mbeukman/xlm-roberta-base-finetuned-ner-swahili

Python, R. (n.d.). *Using fastapi to build python web apis – real python*. Retrieved July 24, 2024, from https://realpython.com/fastapi-python-web-apis/

Spark, C. (2019, December 23). *Hyperparameter tuning in XGBoost*. Medium. https://blog.cambridgespark.com/hyperparameter-tuning-in-xgboost-4ff9100a3b2f

Transformers in machine learning. (2021, March 8). GeeksforGeeks.

https://www.geeksforgeeks.org/getting-started-with-transformers/

What is fine-tuning? | ibm. (2024, March 15). https://www.ibm.com/topics/fine-tuning