B5W4: Building an Amharic E-commerce Data Extractor

Transform messy Telegram posts into a smart FinTech engine that reveals which vendors are the best candidates for a loan.

OVERVIEW

Business Need

EthioMart has a vision to become the primary hub for all Telegram-based e-commerce activities in Ethiopia. With the increasing popularity of Telegram for business transactions, various independent e-commerce channels have emerged, each facilitating its own operations. However, this decentralization presents challenges for both vendors and customers who need to manage multiple channels for product discovery, order placement, and communication.

To solve this problem, **EthioMart** plans to create a single centralized platform that consolidates real-time data from multiple e-commerce Telegram channels into one unified channel. By doing this, they aim to provide a seamless experience for customers to explore and interact with multiple vendors in one place.

This project focuses on fine-tuning **LLM's** for Amharic Named Entity Recognition (NER) system that extracts key business entities such as product names, prices, and Locations, from text, images, and documents shared across these Telegram channels. The extracted data will be used to populate **EthioMart**'s centralised database, making it a comprehensive e-commerce hub.

Key Objectives:

- Develop a repeatable workflow that begins with data ingestion from Telegram channels, proceeds through preprocessing and labeling, and results in structured, machine-readable data.
- Fine-tune a transformer-based model to achieve high accuracy (measured by F1-score) in identifying Product, Price, and Location entities within unstructured Amharic text.
- Go beyond just building a model by comparing multiple approaches, interpreting your model's predictions with tools like SHAP/LIME, and delivering a final analysis that recommends the best model for EthioMart's business case.

Possible entities

- Product Names or Types
- Material or Ingredients: Specific mentions of materials used in the products.
- Location Mentions
- Monetary Values or Prices
- Optional entities to be collected
    - DELIVERY_FEE: To capture transaction costs beyond the product price.
        - *Examples:* "free delivery", "150 birr delivery fee", "delivery cost extra".
    - CONTACT_INFO: To capture the means of completing a transaction.
        - *Examples:* Phone numbers (09...), Telegram usernames (@username).

DATA

Source: Messages and data from Ethiopian-based e-commerce Telegram channels.

- Sample data collected from the Shageronlinestore link
- Amharic news labelled NER data set link

Types:

Text (Amharic language messages)

Images (Product images, marketing materials)

KNOWLEDGE AND SKILLS

1. **Text Processing:** Handling Amharic text, tokenization, and preprocessing techniques.

2. **LLM Fine-tuning:** Adapting large language models for Amharic NER tasks.
3. **Model Comparison & Selection:** Evaluating performance using metrics like F1-score, precision, and recall.
4. **Model Interpretability:** Using tools such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) to explain model predictions and outputs.

LEARNING OUTCOMES

By the end of this challenge, you will have developed and demonstrated the ability to:

- Programmatically collect text and image data from live web sources (like Telegram) and preprocess it for a machine learning workflow.
- Apply standard NER labeling schemes (e.g., CoNLL format) to raw Amharic text, creating a high-quality, bespoke dataset for model fine-tuning.
- Adapt large, pre-trained language models (like XLM-Roberta or mBERT) for a specialized downstream task (Amharic NER) using the Hugging Face ecosystem.
- Systematically evaluate and compare different models based on performance metrics (F1-score, Precision, Recall) and resource constraints to justify a model selection.
- Employ model interpretability techniques (SHAP and LIME) to diagnose model behavior, explain predictions, and build trust in the system's outputs.
- Articulate the connection between a technical solution (an NER model) and a business objective (improving an e-commerce platform) in a clear, professional report.

TEAM

Tutors:

- Mahlet
- Rediet
- Kerod
- Rehmet

KEY DATES

- Discussion on the case - 09:30 UTC on Wednesday 18 June 2025. Use #all-week-4 to pre-ask questions.
- Interim Solution - 20:00 UTC on Sunday 22 June 2025.
- Final Submission - 20:00 UTC on Tuesday 24 June 2025

INSTRUCTIONS

The task is divided into the following objectives

- Amharic data collection and preprocessing
- Labeling amharic data for NER
- Fine-Tuning existing models for NER
- Model comparison with different NER models

Task 1: Data Ingestion and Data Preprocessing

Set up a data ingestion system to fetch messages from multiple Ethiopian-based Telegram e-commerce channels. Prepare the raw data (text, images) for entity extraction.

- List of channels
- You have to select atleast 5 channels to fetch data and you can share each other since Fine-Tuning needs more data

Steps:

1. Identify and connect to relevant Telegram channels using a custom scraper.
2. Implement a message ingestion system to collect text, images, and documents as they are posted in real time.
3. Preprocess text data by tokenizing, normalizing, and handling Amharic-specific linguistic features.
4. Clean and structure the data into a unified format, separating metadata (e.g., sender, timestamp) from message content.
5. Store preprocessed data in a structured format for further analysis.

Task 2 : Label a Subset of Dataset in CoNLL Format

- You are tasked with labeling a portion of the provided dataset in the **CoNLL format**. This format is commonly used for Named Entity Recognition (NER) tasks.

- The goal is to identify and label entities such as products, price, and Location in Amharic text.
- Use the above dataset "Message" column of a larger dataset. Each message consists of text describing various products and entities.
    - **CoNLL Format**:
        - Each token (word) is labeled on its own line.
        - The token is followed by its entity label.
        - Blank lines separate individual sentences/messages.
    - **Entity Types**:
        - **B-Product**: The beginning of a product entity (e.g., "Baby bottle").
        - **I-Product**: Inside a product entity (e.g., the word "bottle" in "Baby bottle").
        - **B-LOC**: The beginning of a location entity (e.g., "Addis abeba", "Bole").
        - **I-LOC**: Inside a location entity (e.g., the word "Abeba" in “Addis abeba”)
        - **B-PRICE**: The beginning of a price entity (e.g., "ዋጋ 1000 ብር", "በ 100 ብር").
        - **I -PRICE**: Inside a price entity (e.g., the word "1000" in “ዋጋ 1000 ብር”)
        - **O**: Tokens that are outside any entities.
    - You need to label at least **30-50 messages** from the provided dataset.
    - Save your work in a plain text file in the **CoNLL format**.

Task 3: Fine Tune NER Model

- **Objective:** Fine-Tune a Named Entity Recognition (NER) model to extract key entities (e.g., products, prices, and location) from Amharic Telegram messages.
- **Steps:**
    1. Use **Google Colab** or any other environment with GPU support for faster training.
    2. Install necessary libraries by running the following commands:
    3. You will use the pre-trained XLM-Roberta **or** bert-tiny-amharic or afroxmlr model, which supports multilingual tasks, including Amharic.
    4. Load the labeled dataset in **CoNLL format** from the previous task.
    5. You can use Hugging Face's datasets library to load the data or manually parse the CoNLL format into a pandas DataFrame.

6. Tokenize the data and align the labels with tokens produced by the tokenizer
7. Set up training arguments, such as learning rate, number of epochs, batch size, and evaluation strategy.
8. Use Hugging Face's Trainer API to fine-tune the model.
9. Evaluate the fine-tuned model on the validation set to check performance.
10. After fine-tuning, save the model for future use.

Task 4: Model Comparison & Selection

- Compare different models and select the best-performing one for the entity extraction task.
- **Steps:**
  1. Fine-Tune multiple models **like XLM-Roberta**: A large multilingual model for NER tasks, or **DistilBERT**: A smaller, lighter model for more efficient NER tasks, or **mBERT** (Multilingual BERT): A multilingual version of BERT, suitable for Amharic or .others?
  2. Evaluate the fine-tuned model on the validation set to check performance.
  3. Compare models based on accuracy, speed, and robustness in handling multi-modal data.
  4. Select the best-performing model for production based on evaluation metrics.

Task 5: Model Interpretability

- Use model interpretability tools to explain how the NER model identifies entities, ensuring transparency and trust in the system.
- **Steps:**
  1. Implement SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) to interpret the model's predictions.
  2. Analyze difficult cases where the model might struggle to identify entities correctly (e.g., ambiguous text, overlapping entities).
  3. Generate reports on how the model makes decisions and identify areas for improvement.

Taks 6: FinTech Vendor Scorecard for Micro-Lending

EthioMart wants to identify its most active and promising vendors to offer them small business loans (micro-lending). A lender can't assess a vendor based on text alone; they need to see evidence of business activity and customer engagement.

Using the data you have scraped, you will combine the entities extracted by your NER model with the metadata available on each Telegram post (like views and timestamps) to create a much richer profile of each vendor.

- Develop a Vendor Analytics Engine:
    - Write a script that processes all the posts from a single vendor and calculates a set of key performance metrics. This engine should score a vendor's potential based on real engagement data.
- Calculate Key Vendor Metrics:
    - For each vendor channel you analyzed, calculate the following:
    - Activity & Consistency:
        - Posting Frequency: The average number of posts per week. (Are they an active business?)
    - Market Reach & Engagement:
        - Average Views per Post: The single best indicator of how many potential customers see their products.
        - Top Performing Post: Identify the post with the highest view count for each vendor. What was the product? What was its price?
    - Business Profile (from your NER model):
        - Average Price Point: The average price of products they list. (Are they a high-volume/low-margin or low-volume/high-margin seller?)
    - Create a Final "Lending Score":
        - Combine these metrics into a simple, weighted "Lending Score" of your own design. For example: Score = (Avg Views * 0.5) + (Posting Frequency * 0.5)

In a separate section of your final report titled " Vendor Scorecard," present a summary table with columns (Avg. Views/Post, Posts/Week, Avg. Price (ETB), Lending Score) comparing the vendors you analyzed across these new metrics.

DELIVERABLES

Interim Submission

- Link to your GitHub code that shows the work done for task-1 and task-2
- The EthioMart higher officials would like to assess your progress on the project. Please provide a data summary that includes your data preparation and labeling steps. This summary should be 1-2 pages in length and must be submitted in PDF format.

Feedback

You may not receive detailed comments on your interim submission but will receive a grade.

Final Submission

- Please prepare a PDF suitable for submission as a blog that outlines your process and exploration results. Focus particularly on data, how you selected specific models, and discuss their performance on the Named Entity Recognition (NER) task after fine-tuning.
- Link to your Github code,

Feedback

You will receive comments/feedback in addition to a grade.

REFERENCE

Fine-tuning NER Models:

- How to fine tune BERT
- Hugging Face Blog on Token Classification
- Roberta Multilingual NER
- NER Datasets from Hugging Face
- How to fine tune amharic models

SHAP (SHapley Additive exPlanations)

- SHAP Official Documentation
- Tutorial on SHAP
- How SHAP works

LIME (Local Interpretable Model-Agnostic Explanations)

- LIME Official GitHub Repository
- LIME for Text Models
- A Guide to Model Interpretability with SHAP and LIME

Alternative free GPU

- papaerspace GPU
- Amazon sagemaker